Volume

5

ORACLE CORPORATION

RMU SHOW STATISTIC DBA HANDBOOK

(Second Edition)

APPENDIX INFORMATION

RICK ANDERSON
ORACLE RDB ENGINEERING
REV. 12/15/98

Table of Contents

Volume 5 Preface

Preface

Oracle Rdb is a general-purpose high-end database management system based on the relational data model. Applications operate in a client-server environment when accessing the database. The RMU Show Statistic utility is used by DBAs to monitor the operation of the application and analyze performance characteristics of the database.

PURPOSE OF THIS HANDBOOK

This handbook describes the various appendix information relevant to the RMU Show Statistic utility. The information contained in this handbook is an amalgamation of various sources of information on the utility, combined into a common frame of reference.

This handbook is not intended to be a reference manual nor should the handbook be construed as a tutorial on how to use the utility. Rather, the information contained in this handbook should be useful for anyone interested in detecting, analyzing and correcting performance problems with the database or application.

INTENDED AUDIENCE

This handbook is intended for experienced database administrators familiar with the RMU Show Statistic utility. You should also be familiar with database management procedures and terminology.

HANDBOOK CONTENTS

The RMU Show Statistic "DBA Handbook" is comprised of five volumes. These are the following:

- Volume 1 - Methods and Internals.
- Volume 2 - Screen Chapters 1 through 7.

- Volume 3 - Screen Chapters 8 Through 13.
- Volume 4 - Screen Chapters 14 Through 22.
- Volume 5 - Appendix Information.

ORACLE CODASYL DBMS

This handbook discusses the Oracle Rdb product. However, the RMU Show Statistic utility is almost identical for the Oracle CODASYL DBMS product DBO Show Statistic utility; in fact, it uses a common same source code. Therefore, the few statistic screens that are unique to the Oracle DBMS product are included in this handbook.

OPERATING SYSTEM INFORMATION

The information contained in this handbook is, for the most part, operating system independent. However, on the rare occasion when an operating system specific item is discussed, the section will be appropriately high-lighted.

RELATED MANUALS

For more information on the RMU Show Statistic utility, and database performance tuning information in general, please refer to the following Oracle Rdb documentation:

- Oracle Rdb RMU Reference Manual
- Oracle Rdb Guide to Database Maintenance
- Oracle Rdb Guide to Database Performance and Tuning

In addition, there are several "white papers" discussing RMU Show Statistic issues available in **MetaLink**. For information on the MetaLink system, please contact Oracle Client Relations or visit the Oracle Support webpage at

http://www.oracle.com/support/

ACKNOWLEDGMENTS

A special word of thanks is extended to Simon Pickering for writing the initial version of the Handbook back in 1995, upon which this new version is loosely based.

A word of thanks is also extended to Kathy Oakey, Bill Gettys, Lilian Hobbs and all the other Oracle Rdb consultants who begged and pleaded for more detailed performance analysis screens to make their job easier. Without these suggestions and ideas, the RMU Show Statistic utility would not be as powerful as it is today.

Finally, I have to thank Anna Logan, who wrote the *Guide to Database Performance and Tuning* manual. She diligently kept after me to keep the "help" documentation up-to-date, even when I was adding screens to the RMU Show Statistic was utility on a daily basis.

AUTHOR

If you have any comments, suggestions or corrections for this handbook, please forward them to the author at the following email address:

rjanders@us.oracle.com

Appendix A

Statistic Identifiers

Appendix A - Statistic Identifiers

The following table lists the statistic identifiers to use when manually creating a custom screen. It is advisable to use the "Yank" onscreen-menu option instead of the manual-entry method.

Id	Statistic Name	Description
1	TRANS	transactions completed
2	VERB_SUCCESS	total verbs executed
3	VERB_FAILURE	total verbs rolled-back
4	PSII_BAL	B-tree index node Balances
5	PSII_CRE	B-tree index node Creations
6	PSII_DES	B-tree index node Destructions
7	PSII_INS	B-tree index Insertions (all)
8	PSII_MOD	B-tree index Modifications
9	PSII_REM	B-tree index Removals (all)
10	PSII_SEA	B-tree index Searches
11	PSII_DIST1	One way scroll distribution
12	PSII_DIST2	Two way scroll distribution

Id	Statistic Name	Description
13	PSII_DIST3	Three way scroll distribution
14	GET_VM	number of calls to GET_VM
15	FREE_VM	number of calls to FREE_VM
16	GET_VM_TOTAL	total bytes of VM received
17	FREE_VM_TOTAL	total bytes of VM freed
18	EXPREG	number of calls to $EXPREG
19	SNAP_EXTEND	snap file extensions
20	SNAP_LCK_CONFLICT	did not wait for page lock
21	SNAP_STORE	stored a snapshot record
22	SNAP_PAGE_FULL	candidate snap page too full
23	SNAP_PAGE_USED	fetched an unusable snap page
24	SNAPPER_READ	snapper read a segment
25	SNAPPER_FETCH	snapper fetched a snap page
26	SNAPPER_FOUND	snapper found target segment
27	CKPT_TOTAL	all checkpoints (checkpoints)
28	CKPT_TRANS	checkpoints due to transaction limit
29	CKPT_AIJ	checkpoints due to after-image journal growth
30	AIJ_NET_BLK_RECV	AIJ Log Server ("ALS") network blocks sent
31	CKPT_DELTA_TRANS	transactions between checkpoints
32	CKPT_DELTA_AIJ	after-image journal growth between checkpoints
33	RUJ_PUT	total .RUJ file puts
34	CKPT_AIJ BACKUP	checkpoints due to after-image journal backup activity
35	READ_FILE	total UTIO$READ_FILE calls

Id	Statistic Name	Description
36	WRITE_FILE	total UTIO$WRITE_FILE calls
37	CKPT_ROLLBACK	checkpoints due to rollbacks
38	CKPT_TIME	checkpoints due to time limit
39	CKPT_DELTA_TIME	time (seconds) between checkpoints
40	CKPT_GLOBAL	checkpoints due to global force
41	AIJ_NET_MSG_DATA	AIJ Log Server ("ALS") network message data send
42	AIJ_NET_MSG_CTRL	AIJ Log Server ("ALS") network message ctrl count
43	AIJ_NET_MSG_CKPT	AIJ Log Server ("ALS") checkpoint operations
44	AIJ_CACHE_OVERFLOW	after-image journal cache filled to capacity
45	AIJ_NUM_GROUPS	number of after-image journal group commits
46	AIJ_SWITCH_OVER	after-image journal switch-over operations
47	AIJ_CONTROL_IO	commit/roll/open I/O operations
48	AIJ_DATA_IO	after-image journal data record I/O operations
49	AIJ_HIBER_STALL	after-image journal I/O hibernate stall time
50	AIJ_SAVED_IO	saved on a data I/O operation
51	AIJ_ARB_USED	total AIJ Request Blocks ("ARB") utilized
52	AIJ_PUT	after-image journal records written
53	AIJ_NO_FREE_ARB	all AIJ Request Blocks ("ARB") were owned
54	AIJ_FILLER_BYTES	unused space in the after-image journal
55	AIJ_BUILD_LOCK	restore global after-image journal lock value block
56	AIJ_NET_MSG_SEND	AIJ Log Server ("ALS") network message send count
57	AIJ_NET_MSG_RECV	AIJ Log Server ("ALS") network message receive count

Id	Statistic Name	Description
58	AIJ_NET_STALL	AIJ Log Server ("ALS") network stall duration
59	AIJ_NET_BLK_SENT	AIJ Log Server ("ALS") network blocks sent
60	SNAP_TRANS	transactions completed
61	DBR	Database Recovery process ("DBR") invoked for dead process
62	LCKMODE_NOT_VALID	Lock release mode is invalid for process state
63	FAIJBL_CNT	Number of entries currently on the AIJ recovery failed AIJBL queue
64	FAILJBL_TTL	Total number of entries ever on the AIJ recovery failed AIJBL queue
65	PIO_SPAM_UPDATE_CLUMP	SPAM fetched for record clump allocation
66	PIO_SPAM_UPDATE_THRSH	SPAM fetched for record threshold update
67	PIO_SPAM_UPDATE_FIB	SPAM fetched for record Fast Incremental Backup update
68	PIO_SPAM_FETCH_SCAN	SPAM fetched for record scan
69	PIO_SPAM_FETCH_STORE	SPAM fetched for record store
70	PIO_SPAM_FETCH_MODIFY	SPAM fetched for record modify
71	PIO_SPAM_FETCH_ERASE	SPAM fetched for record erase
72	PIO_SPAM_UPDATE_STORE	SPAM updated for record store
73	PIO_SPAM_UPDATE_MODIFY	SPAM updated for record modify
74	PIO_SPAM_UPDATE_ERASE	SPAM updated for record erase
75	PIO_D_TRANSFERRED	Transferred data page (no I/O)
76	PIO_S_TRANSFERRED	Transferred spam page (no I/O)
77	PIO_SVC_BLKAST	Service long-stream blocking AST
78	PIO_BLKAST_SVCD	Long-stream blocking AST serviced
79	PIO_D_FETCH_RET	fetches for retrieval (read) only

Id	Statistic Name	Description
80	PIO_D_FETCH_UPD	fetches for update
81	PIO_D_LB_ALLOK	in local buffer, correct version & lock mode
82	PIO_D_LB_GBNEEDLOCK	in local buffer, need lock cause of global buffer
83	PIO_D_LB_NEEDLOCK	in local buffer, but wrong lock mode
84	PIO_D_LB_OLDVER	in local buffer, but wrong version
85	PIO_D_GB_NEEDLOCK	in global buffer, right version, need lock
86	PIO_D_GB_OLDVER	in global buffer, but wrong version
87	PIO_D_NOTFOUND_IO	not in buffer pool(s), read in
88	PIO_D_NOTFOUND_SYN	not in buffer pool(s), synth.
89	TRANS_LONG_TIME_TOT	total transaction duration for transactions
90	TRANS_TIME_TOT	total transaction durations
91	TRANS_TIME_VEC_0	histogram ...
92	TRANS_TIME_VEC_1	histogram ...
93	TRANS_TIME_VEC_2	histogram ...
94	TRANS_TIME_VEC_3	histogram ...
95	TRANS_TIME_VEC_4	histogram ...
96	TRANS_TIME_VEC_5	histogram ...
97	TRANS_TIME_VEC_6	histogram ...
98	TRANS_TIME_VEC_7	histogram ...
99	TRANS_TIME_VEC_8	histogram ...
100	TRANS_TIME_VEC_9	histogram ...
101	TRANS_TIME_VEC_10	histogram ...
102	TRANS_TIME_VEC_11	histogram ...
103	TRANS_TIME_VEC_12	histogram ...

Id	Statistic Name	Description
104	TRANS_TIME_VEC_13	histogram ...
105	TRANS_TIME_VEC_14	histogram ...
106	TRANS_TIME_VEC_15	histogram ...
107	TRANS_TIME_VEC_16	histogram ...
108	TRANS_TIME_VEC_17	histogram ...
109	TRANS_TIME_VEC_18	histogram ...
110	TRANS_TIME_VEC_19	histogram ...
111	TRANS_TIME_VEC_20	histogram ...
112	PIO_S_FETCH_RET	fetches for retrieval (read) only
113	PIO_S_FETCH_UPD	fetches for update
114	PIO_S_LB_ALLOK	in local buffer, correct version & mode
115	PIO_S_LB_GBNEEDLOCK	in local buffer, need lock due to global buffer
116	PIO_S_LB_NEEDLOCK	in local buffer, wrong lock mode
117	PIO_S_LB_OLDVER	in local buffer, but wrong version
118	PIO_S_GB_NEEDLOCK	in global buffer, right version, need lock
119	PIO_S_GB_OLDVER	in global buffer, but wrong version
120	PIO_S_NOTFOUND_IO	not in buffer pool(s), read in
121	PIO_S_NOTFOUND_SYN	not in buffer pool(s), synth.
122	RUJ_CACHE_OVERFLOW	RUJ cache filled to capacity
123	RUJ_CONTROL_IO	RUJ commit/roll/open I/O operations
124	RUJ_DATA_IO	RUJ data record I/O operations
125	TSNBLK_REFRESH_SAVED	TSNBLK lock refresh saved
126	TSNBLK_LOCK_SAVED	TSNBLK lock saved
127	TSNBLK_LOCK_NEEDED	TSNBLK lock needed

Id	Statistic Name	Description
128-257	CLIENT_STATS	client statistics block
258	RCCH_LATCH_RQSTS	latch requests
259	RCCH_LATCH_RETRIED	latch retried
260	RCCH_SEARCH_CACHE	cache searches
261	RCCH_FOUND_IN_WS	found in working set
262	RCCH_FOUND_IN_CACHE	found in cache
263	RCCH_FOUND_TOO_BIG	found - record too big
264	RCCH_INSERT_CACHE	insert into cache
265	RCCH_INSERT_TOO_BIG	insert - record too big
266	RCCH_INSERT_CACHE_FULL	insert - cache full
267	RCCH_INSERT_COLLISION	insert - hash collision
268	RCCH_VLM_RQSTS	VLM record requests
269	RCCH_VLM_TURNS	VLM window turns
270	RCCH_SKIP_MARKED_SLOTS	skipped marked slots
271	RCCH_SKIP_INUSE_SLOTS	skipped inuse slots
272	RCCH_HASH_MISSES	hash table overflow penalty
273	RCCH_UNMARK	unmarked record in cache
274-289	reserved for future use	
290	RCS_FIND_BUFFER_STALL	RCS find-free buffer stall
291	DASHBOARD_UPDATED	User dashboard updated
292	VALBLK_NOT_VALID	Lock Value Block not valid
293	DBR_STALL	Database Recovery process (“DBR”) duration (freeze lock)

Id	Statistic Name	Description
294	DBR_NOUNDO_CNT	Database Recovery process ("DBR") no-UNDO-needed
295	DBR_UNDO_CNT	Database Recovery process ("DBR") UNDO recovery count
296	DBR_UNDO_STALL	Database Recovery process ("DBR") duration (UNDO)
297	DBR_REDO_CNT	Database Recovery process ("DBR") REDO recovery count
298	DBR_REDO_STALL	Database Recovery process ("DBR") duration (REDO)
299	DBR_RCCH_STALL	Database Recovery process ("DBR") duration (Record Cache)
300	DBR_GB_STALL	Database Recovery process ("DBR") duration (Global Buffers)
301	DBR_AIJ_STALL	Database Recovery process ("DBR") duration (AIJ)
302	DBR_COMMIT_CNT	Database Recovery process ("DBR") committed transaction
303	DBR_ROLLBACK_CNT	Database Recovery process ("DBR") duration (AIJ)
304	DBR_NORESOLVE_CNT	Database Recovery process ("DBR") duration (AIJ)
305	AIJ_ARB_PRALLOC	total AIJ Request Blocks ("ARB") pre-allocated
306	RUJ_DBK_CACHE_OVFL	RUJ DBKEY cache overflow
307	RUJ_DBK_CACHE_HITS	RUJ DBKEY cache hits
308	LOGICAL_NAME_TRANS	logical name translation
309	LOGICAL_NAME_DEFLT	logical name existed
310	PIO_READY	Area readied
311	PIO_OPEN_SLOW_CNT	Area file opened slow method

Id	Statistic Name	Description
312	PIO_OPEN_SLOW_STL	Area file opened slow stall
313	PIO_OPEN_FAST_CNT	Area file opened fast method
314	PIO_OPEN_FAST_STL	Area file opened fast stall
315	TRANS_2PC_COUNT	2PC transaction count
316	TRANS_2PC_TIME_TOT	total 2PC transaction durations
317	TRANS_REG_TIME_TOT	total regular transaction durations
318	PREPARE_STALL	Prepared transaction (2PC) duration
319	TRANS_2PC_RESOLVED	2PC transaction resolved by Database Recovery process ("DBR")
320	TRANS_2PC_RES_CMT	2PC transaction committed by Database Recovery process ("DBR")
321	TRANS_2PC_RES_RLBK	2PC transaction aborted by Database Recovery process ("DBR")
322	RCCH_LATCH_GRANTED	RCCH Latch granted
323	RCCH_LATCH_DEMOTED	RCCH Latch demoted
324	RCCH_LATCH_STALLED	RCCH Latch stall count
325	RCCH_LATCH_STL_HSEC	RCCH Latch stall duration
326	RCCH_LATCH_DEADLOCK	RCCH Latch deadlock count
327	TX_COMMIT	Number of transaction commit operations
328	TX_ROLLBACK	Number of transaction rollback operations
329	TX_PREPARED	Number of distributed transaction prepare operations
330	UNDO_STALL	Transaction rollback duration, in hundredths of seconds
331-350	STALL_CNT	Per-class stall count -
351-	STALL_TIME	Per-class stall duration -

Id	Statistic Name	Description
370		
371-390	STALL_ACTIVE	Per-class active stalls -
391	RW_XTRANS_LONG_TIME_TOT	total transaction duration for transactions
392	RW_XTRANS_TIME_TOT	total transaction durations
393-408	RW_XTRANS_TIME_VEC	histogram ... -
409	RO_XTRANS_LONG_TIME_TOT	total transaction duration for transactions
410	RO_XTRANS_TIME_TOT	total transaction durations
411-426	RO_XTRANS_TIME_VEC	histogram ... -
427	XTRANS_LONG_TIME_TOT	total transaction duration for transactions
428	XTRANS_TIME_TOT	total transaction durations
429-444	XTRANS_TIME_VEC	histogram ... -
445	AIJ_SWITCH_SUSPND	after-image journal switch-over suspended
446	AIJ_TX_SPAN	after-image journal "span" size per transaction
447	VERB_UNDO_STALL	Verb rollback duration, in hundredths of seconds
448	PSII_NODE_FET	B-tree index nodes fetched (all)
449	PSII_LEAF_FET	B-tree index leaf nodes fetched
450	PSII_DUP_FET	B-tree index duplicate nodes fetched
451	PSII_SCAN	B-tree index scans
452	PSII_PRIM_ENT	B-tree index primary entries in scan
453	PSII_DUP_ENT	B-tree index dup entries in scan
454	PSII_ROOT_INS	B-tree index root node insertions
455	PSII_LEAF_INS	B-tree index leaf node insertions

Id	Statistic Name	Description
456	PSII_DUP_INS	B-tree index duplicate node insertions
457	PSII_NODE_CRE	B-tree index nodes created (all)
458	PSII_ROOT_SPLIT	B-tree index root node splits
459	PSII_LEAF_CRE	B-tree index leaf nodes created
460	PSII_DUP_CRE	B-tree index duplicate nodes created
461	PSII_ROOT_REM	B-tree index root node removals
462	PSII_LEAF_REM	B-tree index leaf node removals
463	PSII_DUP_REM	B-tree index duplicate node removals
464	PSII_NODE_DEL	B-tree index leaf nodes deleted
465	PSII_LEAF_DEL	B-tree index leaf nodes deleted
466	PSII_DUP_DEL	B-tree index duplicate nodes deleted
467	HASH_INS	Hash index insertion (all)
468	HASH_INS_DUP	Hash index duplicate inserted
469	HASH_DEL	Hash index deletion
470	HASH_DEL_DUP	Hash index duplicate deletion
471	HASH_SCANS	Hash index scans
472	HASH_FETCH	Hash index node fetches (all)
473	HASH_FETCH_FRAG	Hash index bucket fragments fetched
474	HASH_FETCH_DUP	Hash index duplicate nodes fetched
475	DIO_FETCH	records fetched
476	DIO_FETCH_FRAG	fragmented records fetched
477	DIO_STORE	records stored
478	DIO_STORE_FRAG	fragmented records stored
479	DIO_PAGES_CHECKED	number of pages checked

Id	Statistic Name	Description
480	DIO_MARKED	records marked
481	DIO_PAGES_DISCARD	number of pages discarded
482	DIO_ERASE	records erased
483	DIO_ERASE_FRAG	fragmented records erased
484	DIO_PAGES_CHECKED_FREE	number of "free" pages checked
485	UPDATE_FIB	update fast incremental backup map
486	MARK_SNUB	mark spam-not-updated bitmap
487	FIB_UPDATED	fast incremental backup map updated
488	FIB_SPAM_UPDATED	fast incremental backup spam updated
489	RCVR_READY	number of storage areas readied to recovery
490	RCVR_UPDATE_SPAM	number of SPAM pages updated during recovery
491	RCVR_APPLY_AIJBL	number of AIJ records applied during recovery
492	RCVR_ERASE_MIXED	number of mixed-format storage area record erase operations during recovery
493	RCVR_ERASE_UNIFORM	number of uniform-format storage area record erase operations during recovery
494	RCVR_MODIFY_MIXED	number of mixed-format storage area record modify operations during recovery
495	RCVR_MODIFY_UNIFORM	number of uniform-format storage area record modify operations during recovery
496	AIJ_TIMEOUT	after-image journal format: AIJ Request Blocks (“ARB”) timed out
497	AIJ_FREE_FORMAT	after-image journal format: free format
498	AIJ_NO_DATA	after-image journal format: AIJ Request Blocks (“ARB”) exhausted
499	AIJ_IO_COMPLETE	after-image journal format: I/O complete
500	AIJ_MIN_DATA	after-image journal format: minimum data found

Id	Statistic Name	Description
501	AIJ_CTRL_REC	after-image journal format: control record found
502	AIJ_DBR	after-image journal format: Database Recovery process ("DBR") process
503	ALS_HIBER	after-image journal switch-over operations
504	ALS_SWITCH_OVER	after-image journal switch-over operations
505	ALS_NUM_GROUPS	number of after-image journal group commits
506	ALS_TTL_ARB_FMT	AIJ Log Server ("ALS") total number of AIJ Request Blocks ("ARB") formatted
507	ALS_BGD_ARB_FMT	AIJ Log Server ("ALS") number of AIJ Request Blocks ("ARB") formatted background
508	AIJ_NET_MSN_GAP	AIJ Log Server ("ALS") number of Message Sequence Number ("MSN") gaps detected
509	DB_BIND_COUNT	Database bind count
510	DB_BIND_STALL	Database bind duration
511	CKPT_STALL	Checkpoint operation duration
512	FIOVEC_AIJ_SY_READ	after-image journal I/O
513	FIOVEC_AIJ_SY_WRIT	
514	FIOVEC_AIJ_EXTEND	
515	FIOVEC_AIJ_SY_READ_BLK	
516	FIOVEC_AIJ_SY_WRIT_BLK	
517	FIOVEC_AIJ_EXTEND_BLK	
518	FIOVEC_AIJ_SY_READ_STL	
519	FIOVEC_AIJ_SY_WRIT_STL	
520	FIOVEC_AIJ_EXTEND_STL	
521	FIOVEC_AIJ_ASY_READ	
522	FIOVEC_AIJ_ASY_WRIT	

Id	Statistic Name	Description
523	FIOVEC_AIJ_ASY_READ_BLK	
524	FIOVEC_AIJ_ASY_WRIT_BLK	
525	FIOVEC_AIJ_ASY_READ_STL	
526	FIOVEC_AIJ_ASY_WRIT_STL	
527	FIOVEC_AIJ_PAGES_DISCRD	
528	FIOVEC_DBS_SY_READ	DBS file I/O
529	FIOVEC_DBS_SY_WRIT	
530	FIOVEC_DBS_EXTEND	
531	FIOVEC_DBS_SY_READ_BLK	
532	FIOVEC_DBS_SY_WRIT_BLK	
533	FIOVEC_DBS_EXTEND_BLK	
534	FIOVEC_DBS_SY_READ_STL	
535	FIOVEC_DBS_SY_WRIT_STL	
536	FIOVEC_DBS_EXTEND_STL	
537	FIOVEC_DBS_AS_READ	
538	FIOVEC_DBS_AS_WRIT	
539	FIOVEC_DBS_AS_READ_BLK	
540	FIOVEC_DBS_AS_WRIT_BLK	
541	FIOVEC_DBS_AS_READ_STL	
542	FIOVEC_DBS_AS_WRIT_STL	
543	FIOVEC_DBS_PAGES_DISCRD	
544	FIOVEC_ROO_SY_READ	ROO file I/O
545	FIOVEC_ROO_SY_WRIT	
546	FIOVEC_ROO_EXTEND	

Id	Statistic Name	Description
547	FIOVEC_ROO_SY_READ_BLK	
548	FIOVEC_ROO_SY_WRIT_BLK	
549	FIOVEC_ROO_EXTEND_BLK	
550	FIOVEC_ROO_SY_READ_STL	
551	FIOVEC_ROO_SY_WRIT_STL	
552	FIOVEC_ROO_EXTEND_STL	
553	FIOVEC_ROO_AS_READ	
554	FIOVEC_ROO_AS_WRIT	
555	FIOVEC_ROO_AS_READ_BLK	
556	FIOVEC_ROO_AS_WRIT_BLK	
557	FIOVEC_ROO_AS_READ_STL	
558	FIOVEC_ROO_AS_WRIT_STL	
559	FIOVEC_ROO_PAGES_DISCRD	
560	FIOVEC_RUJ_SY_READ	RUJ file I/O
561	FIOVEC_RUJ_SY_WRIT	
562	FIOVEC_RUJ_EXTEND	
563	FIOVEC_RUJ_SY_READ_BLK	
564	FIOVEC_RUJ_SY_WRIT_BLK	
565	FIOVEC_RUJ_EXTEND_BLK	
566	FIOVEC_RUJ_SY_READ_STL	
567	FIOVEC_RUJ_SY_WRIT_STL	
568	FIOVEC_RUJ_EXTEND_STL	
569	FIOVEC_RUJ_AS_READ	
570	FIOVEC_RUJ_AS_WRIT	

Id	Statistic Name	Description
571	FIOVEC_RUJ_AS_READ_BLK	
572	FIOVEC_RUJ_AS_WRIT_BLK	
573	FIOVEC_RUJ_AS_READ_STL	
574	FIOVEC_RUJ_AS_WRIT_STL	
575	FIOVEC_RUJ_PAGES_DISCRD	
576	PIO_D_ASYNC_FETCH	Asynchronous Data fetches requests
577	PIO_S_ASYNC_FETCH	Asynchronous Spam fetches requests
578	PIO_D_ASYNC_READ_IO	Asynchronous read I/O-data
579	PIO_S_ASYNC_READ_IO	Asynchronous read I/O-Spam
580	PIO_ASYNC_READ_STALL_C	Asynchronous read stall count
581	PIO_SYNC_BATCH_WRITE	Synchronous Batch Writes
582	PIO_ASYNC_BATCH_WRITE	Asynchronous Batch Writes
583	PIO_ASYNC_WRITE_STALL_C	Asynchronous write stall count
584	LCKVEC_TOTAL_LOCK	total lock statistics
585	LCKVEC_TOTAL_UNLOCK	
586	LCKVEC_TOTAL_PROMOTE	
587	LCKVEC_TOTAL_DEMOTE	
588	LCKVEC_TOTAL_BLAST	
589	LCKVEC_TOTAL_LOCK_NOTQ	
590	LCKVEC_TOTAL_LOCK_STAL	
591	LCKVEC_TOTAL_PROM_NOTQ	
592	LCKVEC_TOTAL_PROM_STAL	
593	LCKVEC_TOTAL_LOCK_DDLK	
594	LCKVEC_TOTAL_PROM_DDLK	

Id	Statistic Name	Description
595	LCKVEC_TOTAL_STALL_HSC	
596	LCKVEC_TOTAL_LOCK_TIME	
597	LCKVEC_TOTAL_PROM_TIME	
598	LCKVEC_STAREA_LOCK	Storage Area locks
599	LCKVEC_STAREA_UNLOCK	
600	LCKVEC_STAREA_PROMOTE	
601	LCKVEC_STAREA_DEMOTE	
602	LCKVEC_STAREA_BLAST	
603	LCKVEC_STAREA_LOCK_NOTQ	
604	LCKVEC_STAREA_LOCK_STAL	
605	LCKVEC_STAREA_PROM_NOTQ	
606	LCKVEC_STAREA_PROM_STAL	
607	LCKVEC_STAREA_LOCK_DDLK	
608	LCKVEC_STAREA_PROM_DDLK	
609	LCKVEC_STAREA_STALL_HSC	
610	LCKVEC_STAREA_LOCK_TIME	
611	LCKVEC_STAREA_PROM_TIME	
612	LCKVEC_PNO_LOCK	Page locks
613	LCKVEC_PNO_UNLOCK	
614	LCKVEC_PNO_PROMOTE	
615	LCKVEC_PNO_DEMOTE	
616	LCKVEC_PNO_BLAST	
617	LCKVEC_PNO_LOCK_NOTQ	
618	LCKVEC_PNO_LOCK_STAL	

Id	Statistic Name	Description
619	LCKVEC_PNO_PROM_NOTQ	
620	LCKVEC_PNO_PROM_STAL	
621	LCKVEC_PNO_LOCK_DDLK	
622	LCKVEC_PNO_PROM_DDLK	
623	LCKVEC_PNO_STALL_HSC	
624	LCKVEC_PNO_LOCK_TIME	
625	LCKVEC_PNO_PROM_TIME	
626	LCKVEC_PLN_LOCK	Page Line Number locks
627	LCKVEC_PLN_UNLOCK	
628	LCKVEC_PLN_PROMOTE	
629	LCKVEC_PLN_DEMOTE	
630	LCKVEC_PLN_BLAST	
631	LCKVEC_PLN_LOCK_NOTQ	
632	LCKVEC_PLN_LOCK_STAL	
633	LCKVEC_PLN_PROM_NOTQ	
634	LCKVEC_PLN_PROM_STAL	
635	LCKVEC_PLN_LOCK_DDLK	
636	LCKVEC_PLN_PROM_DDLK	
637	LCKVEC_PLN_STALL_HSC	
638	LCKVEC_PLN_LOCK_TIME	
639	LCKVEC_PLN_PROM_TIME	
640	LCKVEC_SEQBLK_LOCK	SEQBLK locks
641	LCKVEC_SEQBLK_UNLOCK	
642	LCKVEC_SEQBLK_PROMOTE	

Id	Statistic Name	Description
643	LCKVEC_SEQBLK_DEMOTE	
644	LCKVEC_SEQBLK_BLAST	
645	LCKVEC_SEQBLK_LOCK_NOTQ	
646	LCKVEC_SEQBLK_LOCK_STAL	
647	LCKVEC_SEQBLK_PROM_NOTQ	
648	LCKVEC_SEQBLK_PROM_STAL	
649	LCKVEC_SEQBLK_LOCK_DDLK	
650	LCKVEC_SEQBLK_PROM_DDLK	
651	LCKVEC_SEQBLK_STALL_HSC	
652	LCKVEC_SEQBLK_LOCK_TIME	
653	LCKVEC_SEQBLK_PROM_TIME	
654	LCKVEC_FILID_LOCK	FILID locks
655	LCKVEC_FILID_UNLOCK	
656	LCKVEC_FILID_PROMOTE	
657	LCKVEC_FILID_DEMOTE	
658	LCKVEC_FILID_BLAST	
659	LCKVEC_FILID_LOCK_NOTQ	
660	LCKVEC_FILID_LOCK_STAL	
661	LCKVEC_FILID_PROM_NOTQ	
662	LCKVEC_FILID_PROM_STAL	
663	LCKVEC_FILID_LOCK_DDLK	
664	LCKVEC_FILID_PROM_DDLK	
665	LCKVEC_FILID_STALL_HSC	
666	LCKVEC_FILID_LOCK_TIME	

Id	Statistic Name	Description
667	LCKVEC_FILID_PROM_TIME	
668	LCKVEC_TSNBLK_LOCK	TSNBLK locks
669	LCKVEC_TSNBLK_UNLOCK	
670	LCKVEC_TSNBLK_PROMOTE	
671	LCKVEC_TSNBLK_DEMOTE	
672	LCKVEC_TSNBLK_BLAST	
673	LCKVEC_TSNBLK_LOCK_NOTQ	
674	LCKVEC_TSNBLK_LOCK_STAL	
675	LCKVEC_TSNBLK_PROM_NOTQ	
676	LCKVEC_TSNBLK_PROM_STAL	
677	LCKVEC_TSNBLK_LOCK_DDLK	
678	LCKVEC_TSNBLK_PROM_DDLK	
679	LCKVEC_TSNBLK_STALL_HSC	
680	LCKVEC_TSNBLK_LOCK_TIME	
681	LCKVEC_TSNBLK_PROM_TIME	
682	LCKVEC_RTUPB_LOCK	RTUPB locks
683	LCKVEC_RTUPB_UNLOCK	
684	LCKVEC_RTUPB_PROMOTE	
685	LCKVEC_RTUPB_DEMOTE	
686	LCKVEC_RTUPB_BLAST	
687	LCKVEC_RTUPB_LOCK_NOTQ	
688	LCKVEC_RTUPB_LOCK_STAL	
689	LCKVEC_RTUPB_PROM_NOTQ	
690	LCKVEC_RTUPB_PROM_STAL	

Id	Statistic Name	Description
691	LCKVEC_RTUPB_LOCK_DDLK	
692	LCKVEC_RTUPB_PROM_DDLK	
693	LCKVEC_RTUPB_STALL_HSC	
694	LCKVEC_RTUPB_LOCK_TIME	
695	LCKVEC_RTUPB_PROM_TIME	
696	LCKVEC_ACTIVE_LOCK	ACTIVE bitmap locks
697	LCKVEC_ACTIVE_UNLOCK	
698	LCKVEC_ACTIVE_PROMOTE	
699	LCKVEC_ACTIVE_DEMOTE	
700	LCKVEC_ACTIVE_BLAST	
701	LCKVEC_ACTIVE_LOCK_NOTQ	
702	LCKVEC_ACTIVE_LOCK_STAL	
703	LCKVEC_ACTIVE_PROM_NOTQ	
704	LCKVEC_ACTIVE_PROM_STAL	
705	LCKVEC_ACTIVE_LOCK_DDLK	
706	LCKVEC_ACTIVE_PROM_DDLK	
707	LCKVEC_ACTIVE_STALL_HSC	
708	LCKVEC_ACTIVE_LOCK_TIME	
709	LCKVEC_ACTIVE_PROM_TIME	
710	LCKVEC_MEMBIT_LOCK	MEMBIT locks
711	LCKVEC_MEMBIT_UNLOCK	
712	LCKVEC_MEMBIT_PROMOTE	
713	LCKVEC_MEMBIT_DEMOTE	
714	LCKVEC_MEMBIT_BLAST	

Id	Statistic Name	Description
715	LCKVEC_MEMBIT_LOCK_NOTQ	
716	LCKVEC_MEMBIT_LOCK_STAL	
717	LCKVEC_MEMBIT_PROM_NOTQ	
718	LCKVEC_MEMBIT_PROM_STAL	
719	LCKVEC_MEMBIT_LOCK_DDLK	
720	LCKVEC_MEMBIT_PROM_DDLK	
721	LCKVEC_MEMBIT_STALL_HSC	
722	LCKVEC_MEMBIT_LOCK_TIME	
723	LCKVEC_MEMBIT_PROM_TIME	
724	LCKVEC_AIJ_LOCK	AIJ locks
725	LCKVEC_AIJ_UNLOCK	
726	LCKVEC_AIJ_PROMOTE	
727	LCKVEC_AIJ_DEMOTE	
728	LCKVEC_AIJ_BLAST	
729	LCKVEC_AIJ_LOCK_NOTQ	
730	LCKVEC_AIJ_LOCK_STAL	
731	LCKVEC_AIJ_PROM_NOTQ	
732	LCKVEC_AIJ_PROM_STAL	
733	LCKVEC_AIJ_LOCK_DDLK	
734	LCKVEC_AIJ_PROM_DDLK	
735	LCKVEC_AIJ_STALL_HSC	
736	LCKVEC_AIJ_LOCK_TIME	
737	LCKVEC_AIJ_PROM_TIME	
738	LCKVEC_SAC_LOCK	Snapshot cursor locks

Id	Statistic Name	Description
739	LCKVEC_SAC_UNLOCK	
740	LCKVEC_SAC_PROMOTE	
741	LCKVEC_SAC_DEMOTE	
742	LCKVEC_SAC_BLAST	
743	LCKVEC_SAC_LOCK_NOTQ	
744	LCKVEC_SAC_LOCK_STAL	
745	LCKVEC_SAC_PROM_NOTQ	
746	LCKVEC_SAC_PROM_STAL	
747	LCKVEC_SAC_LOCK_DDLK	
748	LCKVEC_SAC_PROM_DDLK	
749	LCKVEC_SAC_STALL_HSC	
750	LCKVEC_SAC_LOCK_TIME	
751	LCKVEC_SAC_PROM_TIME	
752	LCKVEC_FRZ_LOCK	database freeze locks
753	LCKVEC_FRZ_UNLOCK	
754	LCKVEC_FRZ_PROMOTE	
755	LCKVEC_FRZ_DEMOTE	
756	LCKVEC_FRZ_BLAST	
757	LCKVEC_FRZ_LOCK_NOTQ	
758	LCKVEC_FRZ_LOCK_STAL	
759	LCKVEC_FRZ_PROM_NOTQ	
760	LCKVEC_FRZ_PROM_STAL	
761	LCKVEC_FRZ_LOCK_DDLK	
762	LCKVEC_FRZ_PROM_DDLK	

Id	Statistic Name	Description
763	LCKVEC_FRZ_STALL_HSC	
764	LCKVEC_FRZ_LOCK_TIME	
765	LCKVEC_FRZ_PROM_TIME	
766	LCKVEC_QUIET_LOCK	Quiet-Point locks
767	LCKVEC_QUIET_UNLOCK	
768	LCKVEC_QUIET_PROMOTE	
769	LCKVEC_QUIET_DEMOTE	
770	LCKVEC_QUIET_BLAST	
771	LCKVEC_QUIET_LOCK_NOTQ	
772	LCKVEC_QUIET_LOCK_STAL	
773	LCKVEC_QUIET_PROM_NOTQ	
774	LCKVEC_QUIET_PROM_STAL	
775	LCKVEC_QUIET_LOCK_DDLK	
776	LCKVEC_QUIET_PROM_DDLK	
777	LCKVEC_QUIET_STALL_HSC	
778	LCKVEC_QUIET_LOCK_TIME	
779	LCKVEC_QUIET_PROM_TIME	
780	LCKVEC__LOCK	Client locks[1]
781	LCKVEC__UNLOCK	
782	LCKVEC__PROMOTE	
783	LCKVEC__DEMOTE	
784	LCKVEC__BLAST	

[1] New in Oracle Rdb 7.0.2.0. These fields were previously used by the obsolete GBPT Slot locks.

Id	Statistic Name	Description
785	LCKVEC__LOCK_NOTQ	
786	LCKVEC__LOCK_STAL	
787	LCKVEC__PROM_NOTQ	
788	LCKVEC__PROM_STAL	
789	LCKVEC__LOCK_DDLK	
790	LCKVEC__PROM_DDLK	
791	LCKVEC__STALL_HSC	
792	LCKVEC__LOCK_TIME	
793	LCKVEC__PROM_TIME	
794	LCKVEC_LAREA_LOCK	logical area locks
795	LCKVEC_LAREA_UNLOCK	
796	LCKVEC_LAREA_PROMOTE	
797	LCKVEC_LAREA_DEMOTE	
798	LCKVEC_LAREA_BLAST	
799	LCKVEC_LAREA_LOCK_NOTQ	
800	LCKVEC_LAREA_LOCK_STAL	
801	LCKVEC_LAREA_PROM_NOTQ	
802	LCKVEC_LAREA_PROM_STAL	
803	LCKVEC_LAREA_LOCK_DDLK	
804	LCKVEC_LAREA_PROM_DDLK	
805	LCKVEC_LAREA_STALL_HSC	
806	LCKVEC_LAREA_LOCK_TIME	
807	LCKVEC_LAREA_PROM_TIME	
808	PIO_BUFFER_UNMARK	buffer flushes (including SPAM)

Id	Statistic Name	Description
809	PIO_SPAM_UNMARK	SPAM page buffer flushes only
810	PIO_TRANS_UNMARK	Transaction end
811	PIO_OVERFLOW_UNMARK	Buffer overflow
812	PIO_BLAST_UNMARK	BLAST received
813	PIO_LOCK_QUOTA_UNMARK	Lock quota reached
814	PIO_CONFLICT_UNMARK	Minimize lock conflict
815	PIO_UNBIND_UNMARK	User unbinds from database
816	PIO_BATCH_FAIL_UNMARK	Failed batch update
817	PIO_AREA_LOCK_UNMARK	Area lock mode change
818	PIO_LAREA_CHANGE_UNMARK	logical area create, delete, resize
819	PIO_INCR_BACKUP_UNMARK	Incremental backup (SPAM)
820	PIO_AIJ_IO_FAIL_UNMARK	AIJ I/O operation fail w/ fast commit
821	PIO_SNAP_TRUNC_UNMARK	Snapshot truncation
822	PIO_CHECKPOINT_UNMARK	Checkpoint
823	PIO_AIJ_BACKUP_UNMARK	AIJ backup optimization.
824	FIOVEC_ACE_SY_READ	ACE file I/O
825	FIOVEC_ACE_SY_WRIT	
826	FIOVEC_ACE_EXTEND	
827	FIOVEC_ACE_SY_READ_BLK	
828	FIOVEC_ACE_SY_WRIT_BLK	
829	FIOVEC_ACE_EXTEND_BLK	
830	FIOVEC_ACE_SY_READ_STL	
831	FIOVEC_ACE_SY_WRIT_STL	
832	FIOVEC_ACE_EXTEND_STL	

Id	Statistic Name	Description
833	FIOVEC_ACE_ASY_READ	
834	FIOVEC_ACE_ASY_WRIT	
835	FIOVEC_ACE_ASY_READ_BLK	
836	FIOVEC_ACE_ASY_WRIT_BLK	
837	FIOVEC_ACE_ASY_READ_STL	
838	FIOVEC_ACE_ASY_WRIT_STL	
839	FIOVEC_ACE_PAGES_DISCRD	
840	OBJMAN_KROOT_BIND_RET	KROOT rootfile object
841	OBJMAN_KROOT_BIND_UPD	
842	OBJMAN_KROOT_REFRESH	
843	OBJMAN_KROOT_MARK	
844	OBJMAN_KROOT_UNMARK	
845	OBJMAN_KROOT_UNBIND	
846	OBJMAN_FILID_BIND_RET	FILID rootfile object
847	OBJMAN_FILID_BIND_UPD	
848	OBJMAN_FILID_REFRESH	
849	OBJMAN_FILID_MARK	
850	OBJMAN_FILID_UNMARK	
851	OBJMAN_FILID_UNBIND	
852	OBJMAN_SEQBLK_BIND_RET	SEQBLK rootfile object
853	OBJMAN_SEQBLK_BIND_UPD	
854	OBJMAN_SEQBLK_REFRESH	
855	OBJMAN_SEQBLK_MARK	
856	OBJMAN_SEQBLK_UNMARK	

Id	Statistic Name	Description
857	OBJMAN_SEQBLK_UNBIND	
858	OBJMAN_TSNBLK_BIND_RET	TSNBLK rootfile object
859	OBJMAN_TSNBLK_BIND_UPD	
860	OBJMAN_TSNBLK_REFRESH	
861	OBJMAN_TSNBLK_MARK	
862	OBJMAN_TSNBLK_UNMARK	
863	OBJMAN_TSNBLK_UNBIND	
864	OBJMAN_AIJDB_BIND_RET	AIJDB rootfile object
865	OBJMAN_AIJDB_BIND_UPD	
866	OBJMAN_AIJDB_REFRESH	
867	OBJMAN_AIJDB_MARK	
868	OBJMAN_AIJDB_UNMARK	
869	OBJMAN_AIJDB_UNBIND	
870	OBJMAN_AIJFB_BIND_RET	AIJFB rootfile object
871	OBJMAN_AIJFB_BIND_UPD	
872	OBJMAN_AIJFB_REFRESH	
873	OBJMAN_AIJFB_MARK	
874	OBJMAN_AIJFB_UNMARK	
875	OBJMAN_AIJFB_UNBIND	
876	OBJMAN_RTUPB_BIND_RET	RTUPB rootfile object
877	OBJMAN_RTUPB_BIND_UPD	
878	OBJMAN_RTUPB_REFRESH	
879	OBJMAN_RTUPB_MARK	
880	OBJMAN_RTUPB_UNMARK	

Id	Statistic Name	Description
881	OBJMAN_RTUPB_UNBIND	
882	OBJMAN_ACTIVE_BIND_RET	ACTIVE bitmap rootfile object
883	OBJMAN_ACTIVE_BIND_UPD	
884	OBJMAN_ACTIVE_REFRESH	
885	OBJMAN_ACTIVE_MARK	
886	OBJMAN_ACTIVE_UNMARK	
887	OBJMAN_ACTIVE_UNBIND	
888	OBJMAN_CPT_BIND_RET	CPT rootfile object
889	OBJMAN_CPT_BIND_UPD	
890	OBJMAN_CPT_REFRESH	
891	OBJMAN_CPT_MARK	
892	OBJMAN_CPT_UNMARK	
893	OBJMAN_CPT_UNBIND	
894	OBJMAN_CLIENT_BIND_RET	CLIENT rootfile object
895	OBJMAN_CLIENT_BIND_UPD	
896	OBJMAN_CLIENT_REFRESH	
897	OBJMAN_CLIENT_MARK	
898	OBJMAN_CLIENT_UNMARK	
899	OBJMAN_CLIENT_UNBIND	
900	OBJMAN_TOTAL_BIND_RET	TOTAL rootfile object
901	OBJMAN_TOTAL_BIND_UPD	
902	OBJMAN_TOTAL_REFRESH	
903	OBJMAN_TOTAL_MARK	
904	OBJMAN_TOTAL_UNMARK	

Id	Statistic Name	Description
905	OBJMAN_TOTAL_UNBIND	
906	OBJMAN_RMUCLT_BIND_RET	RMU CLIENT object
907	OBJMAN_RMUCLT_BIND_UPD	
908	OBJMAN_RMUCLT_REFRESH	
909	OBJMAN_RMUCLT_MARK	
910	OBJMAN_RMUCLT_UNMARK	
911	OBJMAN_RMUCLT_UNBIND	
912	PIO_D_DAPF_SUCC	DAPF fetch requests ok
913	PIO_S_DAPF_SUCC	DAPF fetch requests ok
914	PIO_D_DAPF_FAIL	DAPF fetch requests failed
915	PIO_S_DAPF_FAIL	DAPF fetch requests failed
916	PIO_D_DAPF_UTIL	DAPF buffer utilized
917	PIO_S_DAPF_UTIL	DAPF buffer utilized
918	PIO_D_DAPF_DISC	DAPF buffer discarded
919	PIO_S_DAPF_DISC	DAPF buffer discarded
920	LCKVEC_DBS_LOCK	DBS locks
921	LCKVEC_DBS_UNLOCK	
922	LCKVEC_DBS_PROMOTE	
923	LCKVEC_DBS_DEMOTE	
924	LCKVEC_DBS_BLAST	
925	LCKVEC_DBS_LOCK_NOTQ	
926	LCKVEC_DBS_LOCK_STAL	
927	LCKVEC_DBS_PROM_NOTQ	
928	LCKVEC_DBS_PROM_STAL	

Id	Statistic Name	Description
929	LCKVEC_DBS_LOCK_DDLK	
930	LCKVEC_DBS_PROM_DDLK	
931	LCKVEC_DBS_STALL_HSC	
932	LCKVEC_DBS_LOCK_TIME	
933	LCKVEC_DBS_PROM_TIME	
934	OBJMAN_RCACHE_BIND_RET	RCACHE rootfile object
935	OBJMAN_RCACHE_BIND_UPD	
936	OBJMAN_RCACHE_REFRESH	
937	OBJMAN_RCACHE_MARK	
938	OBJMAN_RCACHE_UNMARK	
939	OBJMAN_RCACHE_UNBIND	
940	AIJ_SYNC_COLD_SEND	AIJ Log Server (“ALS”) number of cold sync messages sent
941	AIJ_SYNC_WARM_SEND	AIJ Log Server (“ALS”) number of warm sync messages sent
942	AIJ_SYNC_HOT_SEND	AIJ Log Server (“ALS”) number of hot sync messages sent
943	AIJ_SYNC_COMMIT_SEND	AIJ Log Server (“ALS”) number of commit sync messages sent
944	AIJ_SYNC_COLD_STALL	AIJ Log Server (“ALS”) cold sync stall duration
945	AIJ_SYNC_WARM_STALL	AIJ Log Server (“ALS”) warm sync stall duration
946	AIJ_SYNC_HOT_STALL	AIJ Log Server (“ALS”) hot sync stall duration
947	AIJ_SYNC_COMMIT_STALL	AIJ Log Server (“ALS”) commit sync stall duration
948	AIJ_NET_RECONNECT	AIJ Log Server (“ALS”) network re-connect
949	AIJ_NET_FREE_SEND	AIJ Log Server (“ALS”) network "free" I/O
950	RW_TRANS_LONG_TIME_TOT	total transaction duration for transactions

Id	Statistic Name	Description
951	RW_TRANS_TIME_TOT	total transaction durations
952-972	RW_TRANS_TIME_VEC	histogram ... -
973	RO_TRANS_LONG_TIME_TOT	total transaction duration for transactions
974	RO_TRANS_TIME_TOT	total transaction durations
975-995	RO_TRANS_TIME_VEC	histogram ... -
996	LCKVEC_NOWAIT_LOCK	
997	LCKVEC_NOWAIT_UNLOCK	
998	LCKVEC_NOWAIT_PROMOTE	
999	LCKVEC_NOWAIT_DEMOTE	
1000	LCKVEC_NOWAIT_BLAST	
1001	LCKVEC_NOWAIT_LOCK_NOTQ	
1002	LCKVEC_NOWAIT_LOCK_STAL	
1003	LCKVEC_NOWAIT_PROM_NOTQ	
1004	LCKVEC_NOWAIT_PROM_STAL	
1005	LCKVEC_NOWAIT_LOCK_DDLK	
1006	LCKVEC_NOWAIT_PROM_DDLK	
1007	LCKVEC_NOWAIT_STALL_HSC	
1008	LCKVEC_NOWAIT_LOCK_TIME	
1009	LCKVEC_NOWAIT_PROM_TIME	
1010	AIJ_SHUFFLE_AVERT	AIJ cache shuffle averted
1011	AIJ_NET_MSG_PROC	Number of hot standby network messages processed by the Log Replication Server (LRS) process on the standby database
1012	TX_NOTMARKED	Number of transaction commit operations where

Id	Statistic Name	Description
		the transaction did not modify any AIJ records
1013	TIO_ERASE	Number of temporary table records erased
1014	TIO_FETCH	Number of temporary table records fetched
1015	TIO_MARKED	Number of temporary table records modified
1016	TIO_STORE	Number of temporary table records stored
1017	OBJMAN_CLTSEQ_BIND_RET	CLTSEQ rootfile object [2]
1018	OBJMAN_CLTSEQ_BIND_UPD	
1019	OBJMAN_CLTSEQ_REFRESH	
1020	OBJMAN_CLTSEQ_MARK	
1021	OBJMAN_CLTSEQ_UNMARK	
1022	OBJMAN_CLTSEQ_UNBIND	

[2] Available in Oracle Rdb 8.0.

Appendix B

Configuration Parameters

Appendix B - Configuration Parameters

The following is the complete list of configuration parameters, in alphabetical order. For more information on these variables and their respective semantics, please refer to the "Online Analysis" facility chapter in Volume 1.

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
ACTIVE_USER	String					Specifies the "Active User Stall Messages" screen configuration options. The valid keywords are **ACTUAL** and **ELAPSED**.
AIJ_ARBS_PER_IO	Numeric	2	0		1	Specifies the default online analysis value of AIJ request blocks per AIJ I/O The RDM$BIND_STATS_AIJ_ARBS_PER_IO logical name allows you to override the default value of AIJ request blocks per AIJ I/O. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "AIJ Analysis" screen.
AIJ_BKGRD_ARB_RATIO	Numeric	50	0		1	Specifies the default online analysis value for the background AIJ request block threshold. The RDM$BIND_STATS_AIJ_BKGRD_ARB_RATIO logical name allows you to override the default value for the background

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						AIJ request block threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "AIJ Analysis" screen.
AIJ_BLKS_PER_IO	Numeric	2	0		1	Specifies the default online analysis value of blocks per AIJ I/O. The RDM$BIND_STATS_AIJ_BLKS_PER_IO logical name allows you to override the default value of blocks per AIJ I/O. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "AIJ Analysis" screen.
AIJ_SEC_TO_EXTEND	Numeric	60	0		1	Specifies the default online analysis value of seconds to AIJ extend. The RDM$BIND_STATS_AIJ_SEC_TO_EXTEND logical name allows you to override the default value of seconds to AIJ extend. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "AIJ Analysis" screen.
ALARM	Numeric	0	0		0	Establishes an alarm interval (in seconds) for the Stall Messages screen from the command line. This is useful when you plan to submit the RMU Show Statistic utility as a batch job. This variable supersedes the ALARM=*seconds* qualifier.
AUTO_ACTIVE_DETECT	Boolean	FALSE	FALSE	TRUE		When the AUTO_NODE_DETECT vari-

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						able is set to the value TRUE, specifies whether or not to actively or passively detect new nodes joining the cluster. Active detection may incur an I/O operation per screen refresh. Passive detection relies on other users on the current node to passively refresh the node information.
AUTO_NODE_DETECT	Boolean	FALSE	FALSE	TRUE		Specifies whether or not to automatically detect new nodes joining the cluster. When the utility detects a new node joining the cluster, the cluster statistics menu will be automatically displayed.
AUTO_RECONNECT	Boolean	FALSE	FALSE	TRUE		Specifies whether or not to automatically reconnect to disconnected nodes.
BROADCAST	Boolean	TRUE	FALSE	TRUE		Specifies whether or not to broadcast messages. The Broadcast variable is enabled by default, if broadcasting of certain messages has been enabled with DCL SET BROADCAST. If broadcasting has been disabled with the DCL SET BROADCAST=NONE command, broadcast messages are not displayed, even if you specify the RMU Show Statistic utility with the Broadcast variable. This variable supersedes the [NO]BROADCAST qualifier.
BTR_FETCH_DUP_RATIO	Numeric	15	0		1	Specifies the default online analysis value of the B-tree index duplicate fetch threshold. The RDM$BIND_STATS_BTR_FETCH_DUP_RATIO logical name allows you to override

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						the default value of the B-tree index duplicate fetch threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "Index Analysis" screen.
BTR_LEF_FETCH_RATIO	Numeric	25	0		1	Specifies the default online analysis value of the B-tree index leaf node fetch threshold. The RDM$BIND_STATS_BTR_LEF_FETCH_RATIO logical name allows you to override the default value of the B-tree index leaf node fetch threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "Index Analysis" screen.
CHECKPOINT_TX	String					Specifies the "Checkpoint" screen transaction configuration options. The valid keywords are **ACTUAL** and **ELAPSED**.
CHECKPOINT_ALARM	Numeric	0	0		0	Establishes an alarm interval (in seconds) for the Checkpoint Information screen. This is useful when you plan to submit the RMU Show Statistic utility as a batch job.
CHECKPOINT_SORT	String					Specifies the "Checkpoint" screen sort configuration options. The valid keywords are **OLDEST_CHECKPOINT**, **OLDEST_TRANSACTION** and **OLDEST_QUIET_POINT**.
CLUSTER_NODES	String					Identifies the set of nodes that are to participate in statistics collection for the current session. The node name(s) should be comma-separated. The keyword

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						ALL_OPEN indicates that statistics should be collected from all nodes on which the database is currently open. Note that the **ALL_OPEN** keyword is never automatically generated. This variable supersedes the CLUSTER=*node_list* qualifier.
CONFIRM_EXIT	Boolean	FALSE	FALSE	TRUE		When the CONFIRM_EXIT variable is set to the value TRUE, specifies that the user will confirm exiting from the utility. This variable is only useful when the INTERACTIVE variable is TRUE.
CUSTOM_LINE_*n*	String			18		Specifies the name of the statistic field to the located on line *"n* of" the "Custom Statistics" screen. Statistics may be entered for lines "1" through "36", although the number of lines that can actually be displayed depends on your terminal. The statistic name must be specified exactly as it appears on its home screen, including leading spaces. Duplicate statistics as well as duplicated line numbers are detected. Note that the specified custom statistics fields are not evaluated until after the database has been opened. Opening the database activates the various screens, which determines the set of custom statistic fields that can be specified. Therefore, some custom statistics fields may not always be available, depending on which database attributes (for instance "global buffers) are active.

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
CYCLE	Numeric	0	0		0	Specifies the interval (in seconds) to automatically migrate to the next screen in the current sub-menu. This variable supersedes the CYCLE=*seconds* qualifier.
DASHBOARD_UPDATE	Boolean	DISABLED	DISABLED	ENABLED		Specifies whether or not dashboard updates are permitted, if you have the proper privileges. This variable supersedes the OPTION=[NO]UPDATE qualifier.
DATABASE	Control					This control variable identifies the database from which the configuration file was generated. This variable is for documentation purposes only, and is primarily useful only when the LOG qualifier is specified.
DBKEY_LOG	String			255		Logs the records accessed during a given processing period by the various attached processes. The filespec is the name of the file to which all accessed DBKEYs are logged. When importing a configuration file which specifies a log file, even if the same log file is specified, a new log file will be automatically created. This variable supersedes the DBKEY_LOG=*dbkey_log* qualifier.
DBR_RATIO	Numeric	15	0		1	Specifies the default online analysis value of the Database Recovery process ("DBR") invocation threshold. The RDM$BIND_STATS_DBR_RATIO logical name allows you to override the default value of the Database Recovery process ("DBR") invocation

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "RUJ Analysis" screen.
DEADLOCK_FULL_DISPLAY	Boolean	DISABLED	DISABLED	ENABLED		Specifies whether or not the "Lock Deadlock History" screen is to display all processes or just those with deadlock messages.
DEADLOCK_LOG	String		255			Specifies that lock deadlock message are to be written to the specified file. This can be useful when you notice a great number of lock deadlock messages being generated, but do not have the resources on hand to immediately investigate and resolve the problem. The file generated by the DEADLOCK_LOG variable can be reviewed later so that the problem can be traced and resolved. When importing a configuration file which specifies a log file, even if the same log file is specified, a new log file will be automatically created. This variable supersedes the DEADLOCK_LOG=*deadlock_log* qualifier.
EVENT_DESCRIPTION	Command					This command describes a Show Statistic "event" and either enables a new event or disables an active event. Please refer to the "User-Defined Events" chapter of the handbook for more information.
FULL_BACKUP_INTRVL	Numeric	6	0		0	Specifies the online analysis full database backup threshold. The RDM$BIND_STATS_FULL_BACKUP_INTRVL

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						logical name allows you to override the full database backup threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "Recovery Analysis" screen.
FULL_DISPLAY	Boolean	FALSE	FALSE	TRUE		Directs Oracle RMU to display full information displays for those screens that have the "Brief" and "Full" onscreen-menu options.
GB_IO_SAVED_RATIO	Numeric	85	0		1	Specifies the online analysis global buffer IO-saved default threshold. The RDM$BIND_STATS_GB_IO_SAVED_RATIO logical name allows you to override the global buffer IO-saved default threshold. You can also set the global buffer IO-saved threshold from the configuration sub-menu in the RMU Show Statistic utility "Buffer Analysis" screen.
GB_POOL_HIT_RATIO	Numeric	85	0		1	Specifies the online analysis global buffer pool hit default threshold. The RDM$BIND_STATS_GB_POOL_HIT_RATIO logical name allows you to override the global buffer pool hit default threshold. You can also set the global buffer pool hit threshold from the configuration sub-menu in the RMU Show Statistic utility "Buffer Analysis" screen.
HISTOGRAM	Boolean	FALSE	FALSE	TRUE		Directs Oracle RMU to display the initial statistics screen in the numbers display mode or the graph dis-

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						play mode. The HISTOGRAM variable value TRUE specifies the graph display mode. The HISTOGRAM variable value FALSE specifies the numbers display mode. This variable supersedes the [NO]HISTOGRAM qualifier.
INCLUDE	Command					This command temporarily switches processing to the quoted string value that defines a new configuration file. This configuration file, in turn, may also specify the INCLUDE command to switch to yet another configuration file. When processing of the command completes, processing of the current configuration file continues. Remember that variable definitions are superseded by subsequent references. The INCLUDE variable detects infinite recursion. The INCLUDE variable is not exported.
INQUIRE	Command					This command prompts the user to enter a value for the specified variable. This command is typically used after the PROMPT command to prompt the user to enter a value for certain variables. This command does not work when a configuration file is imported. The INQUIRE variable is not exported.
INTERACTIVE	Boolean	TRUE	FALSE	TRUE		Displays the statistics dynamically to your terminal. The Interactive quali-

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						fier is the default when you execute the RMU Show Statistic utility from a terminal. You can use the NOINTERACTIVE qualifier with the Output qualifier to generate a binary statistics file without generating a terminal display. The NOINTERACTIVE qualifier is the default when you execute the RMU Show Statistic utility from a batch job. Note that most of these variable are not interesting when the INTERACTIVE variable is set to FALSE. This variable supersedes the [NO]INTERACTIVE qualifier.
LB_PAGE_HIT_RATIO	Numeric	75	0		1	Specifies the online analysis LB/AS page hit default threshold. The RDM$BIND_STATS_LB_PAGE_HIT_RATIO logical name allows you to override the LB/AS page hit default threshold. You can also set the local buffer pool hit threshold from the configuration sub-menu in the RMU Show Statistic utility "Buffer Analysis" screen.
LOGICAL_AREA	String					Specifies the "By Logical Area" screen configuration options. The valid keywords are **INDIVIDUAL** and **AGGREGATE**.
LOGICAL_OVERVIEW_STAT[3]	String					Specifies the "Logical Area Overview" screen statistic type configuration options. The valid keywords

[3] Available in Oracle Rdb 7.0.2.0.

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						are **CUR_TOTAL, CUR_RATE, MAX_RATE, AVG_RATE** and **PER_TRANS**.
LOGICAL_OVERVIEW_TYPE[4]	String					Specifies the "Logical Area Overview" screen configuration options. The valid keywords are **TABLE, BTREE, HASH** and **BLOB**.
MAX_HASH_QUE_LEN	Numeric	2	0		0	Specifies the online analysis hash table queue length default threshold. The RDM$BIND_STATS_MAX_HASH_QUE_LEN logical name allows you to override the hash table queue length default threshold. You can also set the hash table queue length threshold from the configuration sub-menu in the RMU Show Statistic utility "Transaction Analysis" screen.
MAX_LOCK_STALL	Numeric	2	0		1	Specifies the online analysis lock stall default threshold. The RDM$BIND_STATS_MAX_LOCK_STALL logical name allows you to override the lock stall default threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "Locking Analysis" screen.
MAX_TX_DURATION	Numeric	15	0		1	Specifies the online analysis transaction duration default threshold. The RDM$BIND_STATS_MAX_TX_DURATION logical name allows you to override the transaction duration default

[4] Available in Oracle Rdb 7.0.2.0.

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						threshold. You can also set the transaction duration threshold from the configuration sub-menu in the RMU Show Statistic utility Transaction Analysis screen.
NOTIFY	String					Notifies the specified system operator or operators when a stall process exceeds the specified alarm interval by issuing a broadcast message and ringing a bell at the terminal receiving the message. The valid operator classes are: CENTRAL, CLUSTER, DISKS, SECURITY, and OPER1 through OPER12. Multiple operator classes can be comma-separated; for example 'OPER11,OPER12'. Be sure to use single-quotes for this variable. This variable supersedes the NOTIFY=oper_classes qualifier.
OUTPUT	String			255		Specifies that the collected statistics are to be written to the specified binary output file. This variable supersedes the OUTPUT=binary_file qualifier.
PAGES_CHECKED_RATIO	Numeric	10	0		1	Specifies the online analysis pages checked default threshold. The RDM$BIND_STATS_PAGES_CHECKED_RATIO logical name allows you to override the pages checked default threshold.. You can also set this threshold from the configuration sub-menu in the

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						RMU Show Statistic utility "Record Analysis" screen.
PROC_ASTLM_RATIO	Numeric	25	0		1	Specifies the online analysis process ASTLM default threshold. The RDM$BIND_STATS_PROC_ASTLM_RATIO logical name allows you to override the process ASTLM default threshold. You can also set the process ASTLM threshold from the configuration sub-menu in the RMU Show Statistic utility "Process Analysis" screen.
PROC_BIOLM_RATIO	Numeric	25	0		1	Specifies the online analysis process BIOLM default threshold. The RDM$BIND_STATS_PROC_BIOLM_RATIO logical name allows you to override the process BIOLM default threshold. You can also set the process BIOLM threshold from the configuration sub-menu in the RMU Show Statistic utility "Process Analysis" screen.
PROC_DIOLM_RATIO	Numeric	25	0		1	Specifies the online analysis process DIOLM default threshold. The RDM$BIND_STATS_PROC_DIOLM_RATIO logical name allows you to override the process DIOLM default threshold. You can also set the process DIOLM threshold from the configuration sub-menu in the RMU Show Statistic utility "Process

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						Analysis" screen.
PROC_ENQLM_RATIO	Numeric	25	0		1	Specifies the online analysis process ENQLM default threshold. The RDM$BIND_STATS_PROC_ENQLM_RATIO logical name allows you to override the process ENQLM default threshold. You can also set the process ENQLM threshold from the configuration sub-menu in the RMU Show Statistic utility "Process Analysis" screen.
PROC_PGFLLM_RATIO	Numeric	25	0		1	Specifies the online analysis process PGFLLM default threshold. The RDM$BIND_STATS_PROC_PGFLLM_RATIO logical name allows you to override the process PGFLLM default threshold. You can also set the process PGFLLM threshold from the configuration sub-menu in the RMU Show Statistic utility "Process Analysis" screen.
PRINT	Command					This command prints the "value" to the log file. This variable is useful if you do not want to use the LOG qualifier, but want to display the value of a variable. This variable is also useful for displaying the "initial" value of a variable before it is changed in the configuration file. Interesting uses of the PRINT command have already been described in this handbook. The PRINT variable is not ex-

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						ported.
PROMPT	Command					This command prints the "value" to the terminal. This variable is typically used in conjunction with the INQUIRE variable, but can also be useful if you do not want to use the LOG qualifier, but want to display the value of a variable. This command does not work when a configuration file is imported. Interesting uses of the PROMPT command have already been described in this handbook. The PROMPT variable is not exported.
RECOVERY_SORT	String					Specifies the "Transaction Recovery Duration Estimate" screen sort configuration options. The valid keywords are **LONGEST_TRANSACTION**, **LONGEST_UNDO**, **LONGEST_REDO** and **LONGEST_FREEZE**.
RECS_FETCHED_RATIO	Numeric	20	0		1	Specifies the online analysis records fetched default threshold. The RDM$BIND_STATS_RECS_FETCHED_RATIO logical name allows you to override the records fetched default threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "Record Analysis" screen.
RECS_STORED_RATIO	Numeric	20	0		1	Specifies the online analysis records

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						stored default threshold. The RDM$BIND_STATS_RECS_STORED_RATIO logical name allows you to override the records stored default threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "Record Analysis" screen.
REDIRECT	Command					This command forces the redirection of another variable's value into itself. This is the means by which variable indirection can be achieved. Command variables and control variables cannot be redirected, but all other variables can be. For example, if the variable HELLO has the value "THERE", and the variable FOO has the value "HELLO", then the command REDIRECT="FOO" causes the variable FOO to have the value "THERE". This command variable is typically used after defining the variable's value using the INQUIRE command. The REDIRECT variable is not exported.
REFRESH_INTERVAL	Numeric	3	0		0	Specifies the statistics collection interval in seconds. If you omit this qualifier, a sample collection is made every 3 seconds. The integer has a normal range of 1 to 180 (1 second to 3 minutes). However, if you specify a negative number for the Time qualifier, the RMU Show Statistic utility interprets the number as hundredths of a second. For exam-

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						ple, REFRESH_INTERVAL="–20" specifies an interval of 20/100 or 1/5th of a second. This variable supersedes the TIME=*seconds* qualifier.
REOPEN_INTERVAL	Numeric	0	0		0	After the specified interval, closes the current output file and opens a new output file without requiring you to exit from the RMU Show Statistic utility. The new output file has the same name as the previous output file, but the version number is incremented by 1. This variable supersedes the REOPEN_INTERVAL=*seconds* qualifier.
REPORT_INTERVAL	Numeric	30	0		0	Specifies the automatic screen capture interval in seconds. If you omit this qualifier, automatic screen capture is disabled.
RESET	Boolean	FALSE	FALSE	TRUE		Specifies whether or not the statistics are to be automatically reset prior to being displayed. This variable is always exported with the value FALSE regardless of its initial value. This variable supersedes the [NO]RESET qualifier.
RUJ_SYNC_IO_RATIO	Numeric	10	0		1	Specifies the online analysis synchronous RUJ I/O default threshold. The RDM$BIND_STATS_RUJ_SYNC_IO_RATIO logical name allows you to override the synchronous RUJ I/O default threshold.

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "RUJ Analysis" screen.
SCREEN	String			255		Specifies that name of the initial screen to be displayed. This variable supersedes the SCREEN=*screen_name* qualifier.
STALL_LOG	String			255		Specifies that stall message are to be written to the specified file. This can be useful when you notice a great number of stall messages being generated, but do not have the resources on hand to immediately investigate and resolve the problem. The file generated by the STALL_LOG variable can be reviewed later so that the problem can be traced and resolved. When importing a configuration file which specifies a log file, even if the same log file is specified, a new log file will be automatically created. This variable supersedes STALL_LOG=*stall_log* qualifier.
STALL_LOG_VERBOSE	Boolean	FALSE	FALSE	TRUE		When the STALL_LOG_VERBOSE variable is set to the value TRUE, specifies that the Stall Messages Log file output will be in "verbose" mode, meaning that all stalled processes will be displayed at each refresh interval, instead of just once per stalled event. This variable supersedes the OPTION=VERBOSE qualifier.
STALL_MESSAGE	String					Specifies the "Stall Messages" screen configuration options. The

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						valid keywords are **ACTUAL** and **ELAPSED**.
SYSTEM_LOGICAL_AREAS	Boolean	TRUE	FALSE	TRUE		Specifies whether or not to display system relations on the "Logical Area" statistics screens. This variable only applies to Oracle Rdb databases; it does not apply to Oracle DBMS databases.
TIMEOUT_FULL_DISPLAY	Boolean	DISABLED	DISABLED	ENABLED		Specifies whether or not the "Lock Timeout History" screen is to display all processes or just those with deadlock messages.
TIMEOUT_LOG	String			255		Specifies that lock timeout message are to be written to the specified file. This can be useful when you notice a great number of lock timeout messages being generated, but do not have the resources on hand to immediately investigate and resolve the problem. The file generated by the TIMEOUT_LOG variable can be reviewed later so that the problem can be traced and resolved. When importing a configuration file which specifies a log file, even if the same log file is specified, a new log file will be automatically created. This variable supersedes the TIMEOUT_LOG=*timeout_log* qualifier.
TX_DISPLAY	String					Specifies the "Transaction Duration" screen display interval. The valid keywords are **MINUTES** for 0-15 minutes and **SECONDS** for 0-15 seconds.
TX_DURATION	String					Specifies the "Transaction Duration" screen configuration options.

Variable Name	Type	Default Value	Min. Value	Max. Value	Scale	Description
						The valid keywords are **TOTAL**, **READ_WRITE** and **READ_ONLY**.
UNSET	Command					This command removes a user defined variable from the symbol table. It is not necessary to unset user defined variables prior to changing their value. It is only necessary to unset a variable if you do not wish it to exist for some reason. Note that using the single-quote is not supported for this command because variables can be specified using double-quotes only. The UNSET variable is not exported.
VERB_SUCCESS_RATIO	Numeric	25	0		1	Specifies the online analysis verb success default threshold. The RDM$BIND_STATS_VERB_SUCCESS_RATIO logical name allows you to override the verb success default threshold. You can also set the verb success threshold from the configuration sub-menu in the RMU Show Statistic utility "Transaction Analysis" screen.

Appendix C

Logical Names

Appendix C - RMU Show Statistic Logical Names

The following is the complete list of logicals used by the RMU Show Statistic utility.

Logical Name	Configuration Variable	Default Value	Description
RDM$BIND_STATS_AIJ_ARBS_PER_IO	RDB_BIND_STATS_AIJ_ARBS_PER_IO	2 blocks	Allows you to override the default value of after-image journal request blocks per after-image journal I/O. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "AIJ Analysis" screen.
RDM$BIND_STATS_AIJ_BKGRD_ARB_RATIO	RDB_BIND_STATS_AIJ_BKGRD_ARB_RATIO	50	Allows you to override the default value for the background after-image journal request block threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "AIJ Analysis" screen.
RDM$BIND_STATS_AIJ_BLKS_PER_IO	RDB_BIND_STATS_AIJ_BLKS_PER_IO	2	Allows you to override the default value of blocks per after-image journal I/O. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "AIJ Analysis" screen.

Logical Name	Configuration Variable	Default Value	Description
RDM$BIND_STATS_AIJ_SEC_TO_EXTEND	RDB_BIND_STATS_AIJ_SEC_TO_EXTEND	60	Allows you to override the default value of seconds to after-image journal extend. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "AIJ Analysis" screen.
RDM$BIND_STATS_BTR_FETCH_DUP_RATIO	RDB_BIND_STATS_BTR_FETCH_DUP_RATIO	15	Allows you to override the default value of the B-tree index duplicate fetch threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "Index Analysis" screen.
RDM$BIND_STATS_BTR_LEF_FETCH_RATIO	RDB_BIND_STATS_BTR_LEF_FETCH_RATIO	25	Allows you to override the default value of the B- tree leaf node fetch threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "Index Analysis" screen.
RDM$BIND_STATS_DBR_RATIO	RDB_BIND_STATS_DBR_RATIO	15	Allows you to override the default value of the Database Recovery process ("DBR") invocation threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "RUJ Analysis" screen.
RDM$BIND_STATS_ENABLED		1	Allows you can disable the writing of database statistics for a process or database. When database statistics are disabled for a process, the RMU Show Statistic utility shows zeros in the fields for each of the display screens

Logical Name	Configuration Variable	Default Value	Description
			for that process. If you want to disable the writing of statistics for all processes on a node, you must define the logical name RDM$BIND_STATS_ENABLED or the configuration parameter RDB_BIND_STATS_ENABLED for each process on that node. By default, the writing of database statistics is enabled for each process on a node; the value is set to 1. Disabling statistics is useful for static, performance-critical applications that have been previously tuned and do not need the information provided by the RMU Show Statistic utility. To disable the writing of database statistics for a process, define the RDM$BIND_STATS_ENABLED logical name or the RDB_BIND_STATS_ENABLED configuration parameter to 0.
RDM$BIND_STATS_FULL_BACKUP_INTRVL	RDB_BIND_STATS_FULL_BACKUP_INTRVL	6	Allows you to override the full database backup threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "Recovery Analysis" screen.
RDM$BIND_STATS_GB_IO_SAVED_RATIO	RDB_BIND_STATS_GB_IO_SAVED_RATIO	85	Allows you to override the GB IO-saved default threshold. You can also set the global buffer IO-saved threshold from the configuration sub-menu in the RMU Show Statistic utility "Buffer Analysis" screen.
RDM$BIND_STATS_GB_POOL_HIT_RATIO	RDB_BIND_STATS_GB_POOL_HIT_RATIO	85	Allows you to override the GB pool hit default threshold.

Logical Name	Configuration Variable	Default Value	Description
			You can also set the global buffer pool hit threshold from the configuration sub-menu in the RMU Show Statistic utility "Buffer Analysis" screen.
RDM$BIND_STATS_LB_PAGE_HIT_RATIO	RDB_BIND_STATS_LB_PAGE_HIT_RATIO	75	Allows you to override the LB/AS page hit default threshold. You can also set the local buffer pool hit threshold from the configuration sub-menu in the RMU Show Statistic utility "Buffer Analysis" screen.
RDM$BIND_STATS_MAX_HASH_QUE_LEN	RDB_BIND_STATS_MAX_HASH_QUE_LEN	2 rows	Allows you to override the hash table queue length default threshold. You can also set the hash table queue length threshold from the configuration sub-menu in the RMU Show Statistic utility "Transaction Analysis" screen.
RDM$BIND_STATS_MAX_LOCK_STALL	RDB_BIND_STATS_MAX_LOCK_STALL	2 seconds	Allows you to override the lock stall default threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "Locking Analysis" screen.
RDM$BIND_STATS_MAX_TX_DURATION	RDB_BIND_STATS_MAX_TX_DURATION	15	Allows you to override the transaction duration default threshold. You can also set the transaction duration threshold from the configuration sub-menu in the RMU Show Statistic utility "Transaction Analysis" screen.
RDM$BIND_STATS_PAGES_CHECKED_RATIO	RDB_BIND_STATS_PAGES_CHECKED_RATIO	10 pages	Allows you to override the pages checked default threshold.

Logical Name	Configuration Variable	Default Value	Description
			You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "Record Analysis" screen.
RDM$BIND_STATS_PROC_ASTLM_RATIO	RDB_BIND_STATS_PROC_ASTLM_RATIO	75	Allows you to override the process ASTLM default threshold. You can also set the process ASTLM threshold from the configuration sub-menu in the RMU Show Statistic utility "Process Analysis" screen.
RDM$BIND_STATS_PROC_BIOLM_RATIO	RDB_BIND_STATS_PROC_BIOLM_RATIO	75	Allows you to override the process BIOLM default threshold. You can also set the process BIOLM threshold from the configuration sub-menu in the RMU Show Statistic utility "Process Analysis" screen.
RDM$BIND_STATS_PROC_DIOLM_RATIO	RDB_BIND_STATS_PROC_DIOLM_RATIO	75	Allows you to override the process DIOLM default threshold. You can also set the process DIOLM threshold from the configuration sub-menu in the RMU Show Statistic utility "Process Analysis" screen.
RDM$BIND_STATS_PROC_ENQLM_RATIO	RDB_BIND_STATS_PROC_ENQLM_RATIO	75	Allows you to override the process ENQLM default threshold. You can also set the process ENQLM threshold from the configuration sub-menu in the RMU Show Statistic utility "Process Analysis" screen.
RDM$BIND_STATS_PROC_PGFLLM_RATIO	RDB_BIND_STATS_PROC_PGFLLM_RATIO	75	Allows you to override the process PGFLLM default threshold. You can also set the process

Logical Name	Configuration Variable	Default Value	Description
			PGFLLM threshold from the configuration sub-menu in the RMU Show Statistic utility "Process Analysis" screen.
RDM$BIND_STATS_RECS_FETCHED_RATIO	RDB_BIND_STATS_RECS_FETCHED_RATIO	20 records	Allows you to override the records fetched default threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "Record Analysis" screen.
RDM$BIND_STATS_RECS_STORED_RATIO	RDB_BIND_STATS_RECS_STORED_RATIO	20 records	Allows you to override the records stored default threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "Record Analysis" screen.
RDM$BIND_STATS_RUJ_SYNC_IO_RATIO	RDB_BIND_STATS_RUJ_SYNC_IO_RATIO	10	Allows you to override the synchronous RUJ I/O default threshold. You can also set this threshold from the configuration sub-menu in the RMU Show Statistic utility "RUJ Analysis" screen.
RDM$BIND_STATS_VERB_SUCCESS_RATIO	RDB_BIND_STATS_VERB_SUCCESS_RATIO	25	Allows you to override the verb success default threshold. You can also set the verb success threshold from the configuration sub-menu in the RMU Show Statistic utility "Transaction Analysis" screen.

Appendix D

Binary Output File Format

Appendix D - Binary Output File Format

The following is a description of the binary output file produced by the RMU Show Statistic utility when the OUTPUT qualifier is specified, or displayed when the INPUT qualifier is specified:

Background Information

Prior to Oracle Rdb v6.1, the binary output files generated by the RMU Show Statistic utility did not contain the "storage area" or "row-cache" information. Consequently, when the binary files were re-played, a valuable portion of the performance information was not available.

Starting with Oracle Rdb v6.1, the RMU Show Statistic utility is able to optionally collect and subsequently display the "storage area" (another name for "by-area" or "by-file") statistics information.

Starting with Oracle Rdb v7.0, the RMU Show Statistic utility is able to optionally collect and subsequently display the "row cache" statistics information.

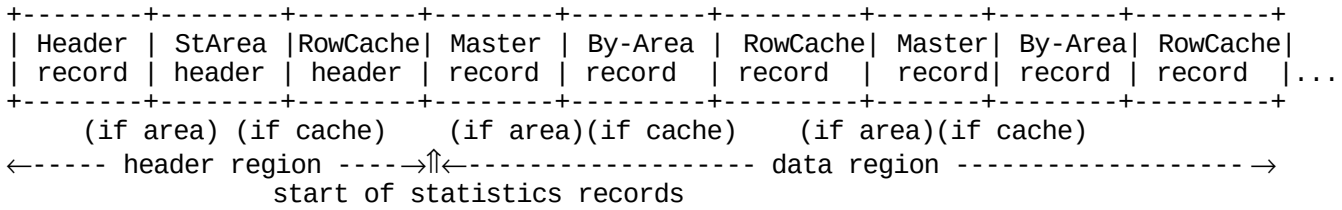
The OPTIONS qualifier is the mechanism by which users select the collection of the by-area statistics. This qualifier is only applicable when used in conjunction with the OUTPUT qualifier; the binary file replay display is always based on what statistics were previously collected.

Binary File Format

With the introduction of the "by-area" statistics collection feature, the binary output file record format has changed from "fixed-length" record format to "variable-length" record format. This change was necessary to support the collection of different size statistics data structures.

The new RMU Show Statistic binary output file now contains two "regions" each containing several different record types.

The binary output file contains the following format:



The binary file regions are:

Header region.

The header region now contains the following record types:

Header record

The header record is a fixed-size record with a length of 4096 bytes. The binary output file header record contains information that describes the unpacking algorithm. The COBOL-style representation of the header record is the following:

```
01 Master_Header_Rec.
05 Date_Time_Stamp   Usage Date.
05 Maj_Ver           Pic X(4).
05 Min_Ver           Pic X(4).
05 Flags             Pic X(4).

05 Arpms_Cnt         Pic X(4).
05 Arpms_Size        Pic X(4).
05 Arpms_Rsz         Pic X(4).
05 Arpms_Rct         Pic X(4).
05 Arpms_Per_Rsz     Pic X(4).

05 Filid_Cnt         Pic X(4).
05 Filid_Size        Pic X(4).
05 Filid_Rsz         Pic X(4).
05 Filid_Rct         Pic X(4).
05 Filid_Per_Rsz     Pic X(4).

05 Rcache_Cnt        Pic X(4).
05 Rcache_Size       Pic X(4).
05 Rcache_Rsz        Pic X(4).
05 Rcache_Rct        Pic X(4).
05 Rcache_Per_Rsz    Pic X(4).

05 Rcpms_Cnt         Pic X(4).
05 Rcpms_Size        Pic X(4).
05 Rcpms_Rsz         Pic X(4).
05 Rcpms_Rct         Pic X(4).
05 Rcpms_Per_Rsz     Pic X(4).
```

```
05 Lapms_Cnt        Pic X(4).
05 Lapms_Size       Pic X(4).
05 Lapms_Rsz        Pic X(4).
05 Lapms_Rct        Pic X(4).
05 Lapms_Per_Rsz    Pic X(4).

05 Max_User_Cnt     Pic X(4).
05 Max_Node_Cnt     Pic X(4).

05 Filler           Pic X(4).
05 Node_Name        Pic X(65).
05 Filler           Pic X(63).
05 Root_File_Name   Pic X(256).
05 Filler           Pic X(3584).
```

"Storage-Area" (FILID) Header Record

The optional storage area header record is a variable-sized record containing a 512-byte structure uniformly packed.

When the RMU Show Statistic utility binary output file contains "by-area" information, a variable number of FILID header records immediately follow the "master" header record. The header record field "Filid_Cnt" identifies the number of FILID header records immediately following the "master" header record. The header record field "Filid_Rsz" identifies the size of each FILID header record. The header record field "Filid_Size" identifies the size in bytes of each FILID entry (currently the value is always "512"), of which there are "Filid_Per_Rsz" in each record.

The COBOL-style representation of the FILID header format is as follows:

```
01 Filid_Header_Rec.
05 Filler           Pic X(12).
05 Flags            Pic X(4).
05 Filler           Pic X(240).
05 Filnam_Len       Pic X(1).
05 Filnam           Pic X(255).
```

“Row Cache” (RCACHE) Header Record

The optional row cache header record is a variable-sized record containing a 512-byte structure uniformly packed.

The COBOL-style representation of the RCACHE header format is as follows:

```
01 Rcache_Header_Rec.
05 Filler           Pic X(4).
05 Flags            Pic X(4).
05 Filler           Pic X(16).
05 Name_Len         Pic X(1).
```

05 Name Pic X(31).
05 Filler Pic X(188).

Data region.

The different record types do not have "tags" identifying them. Each record is placed in a specific sequence, depending on flags and values stored in the header record.

The data region now contains the following record types:

"Master" Statistics Record

Each master statistics record contains one master statistic entry. No unpacking is necessary. The master statistic entry is essentially the "summary" statistic information documented in Appendix A.

The master statistics data record size is 4096 bytes. The format of the master statistics data record is documented in great detail in the "Database Maintenance & Tuning Guide".

"By-Area" Statistics Record

(512-byte structures uniformly packed in variable-size record),

IF by-area information is recorded. Each storage area record actually contains several (one or more) storage area entries, tightly packed. The binary output file header record contains information that describes the unpacking algorithm.

When the RMU Show Statistic utility binary output file contains "by-area" information, a variable number of storage area data ARPMS records immediately follow each master statistics data record. The header record field "Arpms_Cnt" identifies the number of ARPMS data records immediately following the "master" data record. The header record field "Arpms_Rsz" identifies the size of each ARPMS header record. The header record field "Arpms_Size" identifies the size in bytes of each ARPMS entry (currently the value is always "512"), of which there are "Arpms_Per_Rsz" in each record.

The ARPMS header format is as follows:

```
01 Filid_Header_Rec.
05 Sync_Read_Cnt              Pic X(4).
05 Sync_Write_Cnt             Pic X(4).
05 File_Extend_Cnt            Pic X(4).
05 Sync_Read_Blkcnt           Pic X(4).
05 Sync_Write_Blkcnt          Pic X(4).
05 File_Extend_Blkcnt         Pic X(4).
05 Sync_Read_Stall            Pic X(4).
05 Sync_Write_Stall           Pic X(4).
05 File_Extend_Stall          Pic X(4).
05 Async_Read_Cnt             Pic X(4).
05 Async_Write_Cnt            Pic X(4).
05 Async_Read_Blkcnt          Pic X(4).
05 Async_Write_Blkcnt         Pic X(4).
05 Async_Read_Stall           Pic X(4).
05 Async_Write_Stall          Pic X(4).
05 Batch_Stall                Pic X(4).
05 Filler                     Pic X(448).
```

"Row Cache" Statistics Record
(512-byte structures uniformly packed in variable-size record), IF row cache information is recorded.

When the RMU Show Statistic utility binary output file contains "row cache" information, a variable number of row cache data RCPMS records follow each master statistics data record. The header record field "Rcpms_Cnt" identifies the number of RCPMS data records immediately following the "master" data record. The header record field "Rcpms_Rsz" identifies the size of each RCPMS header record. The header record field "Rcpms_Size" identifies the size in bytes of each RCPMS entry (currently the value is always "512"), of which there are "Rcpms_Per_Rsz" in each record.

Data Region Parsing Algorithm

The algorithm for reading the RMU Show Statistic utility binary output file is the following:

I. Read master header record (4096 bytes)

II. Allocate FILID buffer (Filid_Cnt * Filid_Size bytes)

III. Allocate ARPMS buffer (Arpms_Cnt * Arpms_Size bytes)

IV. For I = 1 to Filid_Rct do

  A. Read filid header record (Filid_Rsz bytes)

V. End For

VI. Until EOF do

A. Read master data record (4096 bytes)

B. For I = 1 to Arpms_Rct do

1. Read arpms data record (Arpms_Rsz bytes)

C. End For

D. For I = 1 to Rcpms_Rct do

1. Read rcpms data record (Rcpms_Rsz bytes)

E. End For

VII. End Until

Appendix E

Stall Messages

Appendix E - Stall Messages

The following table describes all of the possible stall messages, in alphabetical order:

Stall Message	Description
binding to database	This message indicates that the process is binding or attaching to the database.
Bugcheck: *name*	This message indicates that the corresponding application process has terminated abnormally and is producing a bugcheck dump for diagnostic analysis.
committing TSN *number : number*	This message indicates that the specified transaction is being committed to the database.
connecting to remote database (*number*)	This message indicates that the AIJ Log Server ("ALS") process, the AIJ Catch-up Server ("LCS"), AIJ Backup Server ("ABS") or Database Recovery server ("DBR") is attaching to the standby database. The *number* value indicates the number of connection attempts remaining before failure.
creating AIJ backup file	This message indicates that the AIJ Backup Server ("ABS") or RMU Backup After_Journal utility is creating an after-image journal backup file.
creating temporary AIJ journal	This message indicates that the AIJ Backup Server ("ABS") or RMU Backup After_Journal utility is creating a temporary after-image journal. This occurs when the Fast Commit feature is enabled, or "no quiet point" after-image journal backups are performed.
extending AIJ file	This message displays whenever the after-image journal is logically or physically extended, which should occur infrequently.
extending .RUJ file	This message displays whenever the .RUJ file is physically extended,

Stall Message	Description
	which should occur infrequently.
extending storage area *number*	This message displays whenever a storage area file (identified by its numeric identifier, which can be determined using the RMU Dump utility) is physically extended. You can determine the numeric identifier for a database's storage areas by using the RMU Dump utility. This message should occur infrequently. However, this message may occur more frequently with WORM areas because WORM area pages cannot be reused once they have been written.
finding buffer for record cache *number*	This message indicates that the specified row cache is being searched for an available buffer.
finding next free buffer	This message indicates that a search for the next available buffer in a row cache is being performed.
hibernating for *number* record cache latch	This message indicates that the latch for the specified row cache could be acquired within a reasonable period of time, so the process will hibernate until the latch is released by the holding process.
hibernating on AIJ I/O completion	This message indicates that the AIJ Log Server ("ALS") is hibernating until the current asynchronous write I/O operation to the after-image journal completes.
hibernating on AIJ submission	This message indicates that the process has submitted after-image journal information and is hibernating while the AIJ Log Server ("ALS") processes the information.
hibernating until next checkpoint (*time*)	This message indicates that the Row Cache Server ("RCS") is hibernating until the next checkpoint, which will occur at the indicated date/time.
Initializing AIJ journal	This message indicates that the current after-image journal is being initialized.
initializing AIJ journal	This message indicates that the after-image journal is being initialized. This occurs when an after-image journal has been backed up.
initializing new AIJ journal	This message indicates that the after-image journal is being initialized. This occurs when an after-image journal is originally created.

Stall Message	Description
latching page *number : number* in GB	This message indicates that the indicated page is being latched for exclusive access in the global buffer pool.
locking page *number : number*	This message indicates that the specified page is being locked.
opening storage area *number* file	This message indicates that the specified storage area file is being opened.
performing cluster statistics collection	This message indicates that the RMU Show Statistic utility is performing statistic collection cluster-wide.
performing local statistics collection	This message indicates the RMU Show Statistic utility is performing statistic collection on the local node only.
prepared, waiting to commit distributed transaction	This message indicates that the process is part of a distributed transaction and that the transaction has been successfully prepared.
Processing pending messages	This message indicates that the standby database log replication server ("LRS") on the standby database is processing messages received from the master database.
Processing pending redo	This message indicates that the standby database log replication server ("LRS") on the standby database is applying the after-image journal information from the master database.
querying standby database for state information	This message indicates that the Log Catch-up Server ("LCS") on the master database is querying the Log Replication Server ("LRS") on the standby database for information about the state of the Hot Standby replication.
RCS waiting for *number* record cache record latch *number*	This message indicates that the row cache server ("RCS") is waiting for the latch for the specified row cache.
reading *number* ACE file blocks from VBN *number*	This message indicates that the database recovery ("DBR") process is recovering the premature termination of the group commit process, and is reading the after-image journal information from the AIJ electronic cache .ACE file.
reading *number* AIJ file blocks from VBN *number*	This message displays whenever the after-image journal lock information needs to be refreshed; this typically only occurs the first time a user attaches to the database. The after-image journal is read to determine the after-image journal logical EOF (not to be confused with the OpenVMS logical EOF). It is also read by the database recovery ("DBR") process.

Stall Message	Description
reading AIJ open record (block 1)	This message indicates the after-image journal "open" record is being read to verify information about the state of the after-image journal.
reading pages *number : number* to *number : number*	This message displays whenever one or more pages is read into either a user's local buffer or the global buffer. One buffer full of pages is being read. The format string "*number : number*" identifies the physical area and the page number.
reading ROOT file (*name* VBN *number*)	This message displays whenever the in-memory database root information is determined to be out-of-date and must be read again from the disk. This message normally occurs only when a database parameter is modified by a user on line or some information in the database root is modified by the system (such as the after-image journal sequence number).
reading .RUJ file block *number*	This message displays whenever an undo operation needs to read the next RUJ page to acquire the rollback information necessary to complete the operation. The .RUJ file is read one block at a time. Sometimes a process that is not being rolled back receives this message because it was necessary to read the .RUJ file in order to refresh cached recovery information.
sending AIJ replication data to standby database	This message indicates that a Hot Standby replication message is being sent from the master database to the standby database.
sending AIJ replication reply to master database	This message indicates that the Log Replication Server ("LRS") on the standby database is sending a reply message to the master database.
switching AIJ journals	This message indicates that an after-image journal switch-over operation is being performed; this message only occurs when circular after-image journals are used.
waiting *number* seconds for next pass through AIJ files	This message indicates that an after-image journal backup operation is suspended for the indicated number of seconds before performing an additional pass through the after-image journals.
waiting for *resource* (*mode*)	This message displays whenever a process requests a lock "with wait" and another process is holding the lock in an incompatible mode. This message may indicate a database hot spot and should be investigated using the RMU Show Locks utility. The format string "*resource*" identifies the lock type (that is, storage area, page, MEMBIT, etc.) and the string "*mode*" identifies the requested lock

Stall Message	Description
	mode (PR, CR, EX, etc.). Appendix F contains a complete list of the locks for this message.
waiting for *number* record cache hash latch	This message indicates that the process is waiting for the specified row cache hash table latch.
waiting for *number* record cache record latch	This message indicates that the process is waiting for the specified row cache record image latch.
waiting for *number*-block unmodified AIJ (*number* minutes)	This message indicates that an after-image journal switch-over operation is suspended, and that an unmodified after-image journal containing at least the indicated number of available blocks is required for the switch-over operation to continue. The "minutes" indicates the time remaining before the database is shutdown.
waiting for active AIJ backups to complete	This message indicates that the process is waiting for active after-image journal backup operations to complete. This message is typically displayed when using the RMU Close utility to close the database.
waiting for AIJ initialization	This message indicates that the process is waiting for an after-image journal initialization operation to complete.
waiting for AIJ message *number* from master database	This message indicates that the process is waiting for a reply from the Log Replication Server ("LRS") on the standby database.
waiting for async-prefetch of pages *number : number* to *number : number*	This message indicates that the process is waiting for the asynchronous pre-fetch (read) of the specified pages to complete. This message may indicate that the "Asynchronous Pre-Fetch" feature attributes may not be appropriate.
waiting for async-write of pages *number : number* to *number : number*	This message indicates that the process is waiting for the asynchronous write of the specified pages to complete. This message may indicate that the "Asynchronous Batch Write" feature attributes may not be appropriate.
waiting for async-write of ROOT file	This message indicates that the process is waiting for the asynchronous write to the database rootfile to complete.
waiting for busy AIJ sequence *number* (*number* minutes)	This message indicates that the specified after-image journal has an active checkpoint held by an active process; this message almost always results during an after-image journal backup operation of circular after-image journals.
waiting for checkpoint completion	This message indicates that a checkpoint operation is being per-

Stall Message	Description
	formed.
waiting for global buffer latch	This message indicates that the process is waiting for the global buffer latch. Note that this latch differs from the global buffer page latch.
waiting for reply from standby database	This message indicates that the process is waiting for a reply from the Log Replication Server ("LRS") on the standby database.
waiting for standby database activity request	This message indicates that the Log Replication Server ("LRS") on the standby database is idle and waiting for messages from the master database.
writing *number* ACE file blocks from VBN *number*	This message displays whenever a group commit process writes the commit information to the after-image journal electronic cache .ACE file. In a high throughput environment, the write buffer length will be as close to 64K as possible.
writing *number* AIJ file blocks from VBN *number*	This message displays whenever a group commit process writes the commit information to the after-image journal. In a high throughput environment, the write buffer length will be as close to 64K as possible.
writing *number* pages back to database	This message displays whenever one or more data pages is written to the database. This is typically caused by a request to access those pages from another process or by detaching from the database.
writing AIJ sequence *number* block *number*	This message displays whenever a group commit process writes the commit information to the after-image journal. In a high throughput environment, the write buffer length will be as close to 64K as possible.
writing ROOT file (*name* VBN *number*)	This message displays whenever the in-memory database root information is modified by a user on line or some information in the database root is modified by the system (such as the after-image journal sequence number).
writing .RUJ file block *number*	This message displays whenever a user process writes data page modification information to the .RUJ file. This message always precedes the next message

Appendix F

Waiting For Stall Types

Appendix F - Stall Types

waiting for record

The DBKEYs in the "Waiting for record" messages are logical DBKEYs. For example:

```
2040037A:2   16:13:36.78  waiting for record 1:0:-4 (CR)
202003A4:5   16:25:18.51  waiting for record 91:155:-1 (CW)
```

In the first line of the example, 2040037A:2 is the process ID, and 16:13:36.78 is the time. The "XX:YY:ZZ" format represents the DBKEY, and you can usually interpret it as "logical area number : page number : line number". However, only positive numbers represent the line number.

When a negative number appears, it refers to the record Adjustable Lock Granularity ("ALG") locking level. By default, there are three page locking levels and the negative numbers are interpreted as follows:

- -4 indicates the complete logical area
- -3 normally indicates 1000 database pages range
- -2 normally indicates 100 database pages range
- -1 normally indicates 10 database pages range

For example, in the second line of the example, the DBKEY occurs in logical area 91 in a range of 10 database pages, one of which is page 155.

waiting for page

The DBKEYs in "Waiting for page" messages are physical DBKEYs; for example:

```
202004AA:1   16:14:20.15  waiting for page 1:727 (PW)
```

This DBKEY format is interpreted as "physical area number : page number".

When a DBKEY provides a physical area number and you want to get the physical area name for the area, issue the RMU Dump Header utility. Then look up the physical area number in the command output to find the name of the physical area.

waiting for SEQBLK

This message is displayed when a transaction starts, commits or aborts and one or more transaction sequence numbers ("TSN") must be allocated. This message is more prevalent when dynamic lock remastering is enabled and a large number of very short transactions are used.

This problem can often be remedied by disabling dynamic lock remastering.

waiting for dbkey scope

This message is displayed when a database user who attached using the DBKEY SCOPE IS TRANSACTION clause has a read/write transaction in progress (giving the user the database key scope lock in CW mode), and a second user who specifies the DBKEY SCOPE IS ATTACH clause (which would give the user the database key scope lock in PR mode) tries to attach. In this situation, the second user's process stalls until the first user's transaction completes.

You can specify the database key scope at run time using the DBKEY SCOPE IS clause of the SQL ATTACH statement. If the DBKEY SCOPE IS clause is used with the SQL CREATE DATABASE or SQL IMPORT statements, the setting is in effect only for the duration of the session of the user who issued the statement; the setting does not become a database root file parameter.

waiting for snapshot cursor

This message displays when a process tries to start a read-only transaction when snapshots are deferred, there is no current read-only transaction, and a read/write transaction is active.

Waiting for snapshot cursor is a normal state if snapshots are deferred. The waiting will end when all read/write transactions started before the first read-only transaction have finished.

waiting for MEMBIT lock

For each database, a membership data structure is maintained. The membership data structure keeps track of the nodes that are accessing the database at any given time. The membership data structure for a database is updated when the first user process from a node attaches to the database and when the last user process from a node detaches from the database.

The "waiting for MEMBIT lock" message means that a process is stalled because the database's membership data structure is in the process of being updated.

waiting for nowait signal

NOWAIT transactions do not wait for locks. If a lock requested by a NOWAIT transaction cannot be granted immediately, Oracle Rdb issues an error message and the transaction aborts. As part of carry-over lock optimization, a NOWAIT transaction requests, acquires, and holds a NOWAIT lock. This signals other processes accessing the database that a NOWAIT transaction exists and results in the release of all carry-over locks. If carry-over locks were not released, a NOWAIT transaction could not access an area held by a WAIT transaction's carry-over lock until the WAIT transaction's process detached from the database.

Every NOWAIT transaction requests the NOWAIT lock at transaction start in CW mode and waits until this lock is granted. When a transaction acquires the NOWAIT lock in CW mode, this indicates that all other users know that a NOWAIT transaction is running and indicates that all carry-over locks have been released, thus reducing the possibility of lock contention.

All transactions request the NOWAIT lock in PR mode at commit time.

- If the NOWAIT lock is granted in PR mode, it indicates that there are no NOWAIT transactions attached to the database and carry-over locks are permitted.

- If the NOWAIT lock request is not granted (because a NOWAIT transaction holds the lock in CW mode), carry-over locks are not permitted.

However, a NOWAIT transaction can experience a delay in acquiring the NOWAIT lock if another transaction is holding the lock. This can result in the following RMU Show Statistic utility stall message:

"waiting for NOWAIT signal (CW)"

If NOWAIT transactions are noticeably slow in executing, you can disable carry-over lock optimization by using the CARRY OVER LOCKS ARE [ENABLED | DISABLED] clause with either the SQL CREATE DATABASE or SQL ALTER DATABASE statements. By default, carry-over locks are enabled.

waiting for client lock

A client lock indicates that an Rdb metadata lock is in use. The term client indicates that Rdb is a client of the Rdb locking services. The metadata locks are used to guarantee memory copies of the metadata (table, index and column definitions) are consistent with the on-disk versions.

The "waiting for client lock" message means the database user is requesting an incompatible locking mode. For example, when trying to drop a table which is in use, the drop operation requests a PROTECTED WRITE lock on the metadata object (such as a table)which is incompatible with the existing PROTECTED READ lock currently used by others of the table.

These metadata locks consist of three longwords. The lock is displayed in text format first, followed by its hexadecimal representation. The text version masks out non-printable characters with a dot (.).

The leftmost value seen in the hexadecimal output contains the identifier of the object. The identifier is described below for tables, routines, modules and storage map areas.

- For tables and views, the id represents the unique value found in the RDB$RELATION ID column of the RDB$RELATIONS system table for the given table.

- For routines, the id represents the unique value found in the RDB$ROUTINE ID column of the RDB$ROUTINES system table for the given routine.

- For modules, the id represents the unique value found in the RDB$MODULE ID column of the RDB$MODULES system table for the given module.

- For storage map areas, the id presents the physical area id. The "waiting for client lock" message on storage map areas is very rare. This may be raised for databases which have been converted from versions prior to Rdb 5. 1.

The next value displayed signifies the object type. The following list describes the client objects and their hexadecimal type values.

- Tables or views 00000004
- Routines 00000006
- Modules 00000015
- Storage map areas 0000000E

The last value in the hexadecimal output represents the lock type. The value 55 indicates this is a client lock.

Note: Because the full client lock output is long, it may require more space than is allotted for the Stall. reason column and therefore can be overwritten by the Lock. ID. column output.

For more detailed lock information on the client lock, use the "LockID" onscreen-menu option.

Appendix G

DBKEY Identification

Appendix G - DBKEY Identification

Logical DBKEYs

The DBKEYs in "waiting for record" stall messages are logical DBKEYs. For example:

waiting for record 1:0:-4 (CR)
waiting for record 91:155:-1 (CW)

In this example of the "waiting for record" message, the first two fields of the "waiting for record" message are not shown. The first field of a "waiting for record" message is the process ID of the stalled process, and the second field is the time the stall began. The third field in the "waiting for record" message (the field with the "XX:YY:ZZ" format) represents the DBKEY, and you can usually interpret it as

"logical area number: : page number : line number. "

However, only positive numbers represent the line number. When a negative number appears, it refers to the record ALG (adjustable lock granularity) locking level. By default, there are three page locking levels and the negative numbers are interpreted as follows:

- -4 indicates the complete logical area
- -3 normally indicates 1000 database pages range
- -2 normally indicates 100 database pages range
- -1 normally indicates 10 database pages range

Of course, your particular database may have more or less than the default three locking levels.

For example, in the second line of the example, the DBKEY occurs in logical area 91in a range of 10 database pages, one of which is page 155.

When you have a logical area number and want to get the physical area name for that logical area, follow these steps:

1. Issue the following command:

   $ RMU /DUMP /LAREAS=RDB$AIP db-name

2. Search the resulting dump for the logical area with that number.

3. Note the corresponding physical area number.

4. Issue the following command:

   $RMU/DUMP/HEADER db-name

5. Look up the physical area number from the output of the RMU Dump Header utility to find the name of the physical area.

You can also look up columns RDB$STORAGE ID or RDB$INDEX ID in system relations RDB$RELATIONS, RDB$INDICES and RDBVMS$STORAGE MAP AREAS to identify the Oracle Rdb entity (table or index) that the DBKEY represents. For a description of the system relations, see the System Relations help topic by issuing the following command:

```
$ HELP SQL SYSTEM RELATIONS
```

The page number field in the DBKEY is the number of the page in the corresponding physical area; the line number is the number of the record on that page.

Physical DBKEYs

The DBKEYs in "waiting for page" stall messages are physical DBKEYs. For example:

```
"waiting for page 1:727 (PW)"
```

In this example of the "waiting for page" message, the first two fields of the "waiting for page" message are not shown. The first field of a "waiting for page" message is the process ID of the stalled process, and the second field is the time the stall began. The DBKEY format for a "waiting for page" message is interpreted as

```
"physical area number: page number. "
```

When you have a physical area number and want to get the physical area name for the area, use the RMU DUMP utility. Then look up the physical area number in the command output to find the name of the physical area.

You can also get a conversion table by issuing the following command:

```
$ RMU /ANALYZE /LAREAS /OPTION=DEBUG /END=1 -
$         /OUTPUT=LAREA. LIS database-name
```

This utility produces a printable file containing all logical areas, logical identifier numbers and physical identifier numbers for a database.

The CR, CW, and PW in the previous examples are requested lock modes of "Concurrent Read", "Concurrent Write", and "Protected Write". The following table shows the lock compatibility between a current transaction and access modes other transactions can specify:

		Mode of Requested Lock				
		SR	SW	PR	PW	EX
Mode	SR	Y	Y	Y	Y	N
of	SW	Y	Y	N	N	N
Cur-rent	PR	Y	N	Y	N	N
Lock	PW	Y	N	N	N	N
	EX	N	N	N	N	N

Key to lock modes:

SR - Shared Read
SW - Shared Write
PR - Protected Read
PW - Protected Write
EX - Exclusive
Y - Locks are compatible
N - Locks are not compatible

Shared Read (SR) and Shared Write (SW) in the table are equivalent to Concurrent Read (CR) and Concurrent Write (CW).

Appendix H

Monitor Log Messages

Appendix H - Monitor Log Messages

Monitor Log Message	Description
number global buffers allocated to recovery process, *number* free out of *number*	identifies the number of global buffers actually allocated to the database recover process ("DBR") and the number of available buffers remaining
number global buffers allocated to user, *number* free out of *number*	identifies the number of global buffers actually allocated to the database process and the number of available buffers remaining
number recovery process(es) still active	identifies the number of database recovery processes ("DBR") still actively recovering the database
number recovery processes started	identifies the number of database recovery processes ("DBR") started
number recovery processes yet to complete	identifies the number of database recovery processes ("DBR") remaining to recover the database completely
abnormal AIJ Backup Server process termination detected	indicates that the AIJ Backup Server ("ABS") failed
abnormal AIJ Backup Server termination detected	indicates that the AIJ Backup Server ("ABS") failed
abnormal AIJ Log Catch-Up Server process termination detected	indicates that the Log Catch-up Server ("LCS") failed
abnormal AIJ Log Catch-Up Server termination detected	indicates that the Log Catch-up Server ("LCS") failed
abnormal AIJ Log Roll-Forward Server process ter-	indicates that the Log Roll-forward Server ("LRS")

Monitor Log Message	Description
mination detected	failed
abnormal AIJ Log Roll-Forward Server termination detected	indicates that the Log Roll-forward Server ("LRS") failed
abnormal AIJ Log Server process termination detected	indicates that the AIJ Log Server ("ALS") failed
abnormal AIJ Log Server termination detected	indicates that the AIJ Log Server ("ALS") failed
abnormal Record Cache Server process termination detected	indicates that the Row Cache Server ("RCS") failed
abnormal Record Cache Server termination detected	indicates that the Row Cache Server ("RCS") failed
abnormal server process termination detected	indicates that the generic server process failed
abnormal user termination detected	indicates that the user process failed
AIJ backup server is backing up AIJ sequence *number*	identifies the after-image journal being backed up by the AIJ Backup Server ("ABS")
All snapshots are *keyword*	indicates that all snapshots are either "allowed" or "not allowed"
All snapshots are *keyword*	indicates that all snapshots are either "disabled" or "not disabled"
CLOSE lock request completed for *resource*	indicates that the database close was successfully acquired
cluster recovery completed successfully	indicates that cluster recovery for a remote node failure completed successfully
cluster recovery waiting for MEMBIT lock	indicates that cluster recovery is waiting for a monitor on another node to respond
cluster recovery waiting for RTUPB lock	indicates that cluster recovery is waiting for access to the list of failed processes
cluster watcher detected invalid MEMBIT lock value block	indicates that a monitor on a remote node failed
cluster watcher is active	indicates that the monitor is actively participating in the cluster recovery mechanism

Monitor Log Message	Description
cluster watcher waiting for MEMBIT lock	indicates that the monitor on this node is waiting to join the cluster recovery mechanism
continuing shutdown of database "*name*"	indicates that shutdown for the specified database is able to proceed
dashboard *name* initialized to value *number*	identifies a dashboard item that has been initialized to the specified value
database "*name*"	identifies the database
database dashboard installed	indicates that the database dashboard has been successfully installed for a particular database
database global section initialized	indicates that the global section has been successfully created for the particular database
database monitor created AIJ backup server *name*	identifies the AIJ Backup Server ("ABS") created by the monitor
database monitor created AIJ log server *name*	identifies the AIJ Log Server ("ALS") created by the monitor
database monitor created Record Cache Server *name*	identifies the Row Cache Server ("RCS") created by the monitor
database monitor created recovery process *name*	identifies the database recovery process ("DBR") created by the monitor
database shutdown waiting for active users to terminate	indicates that database shutdown is waiting for active users to terminate; usually in response to a "FORCEX" or "NOABORT" shutdown
database shutdown waiting for recovery to terminate	indicates that database shutdown is waiting for recovery processes to complete; usually in response to a "DELPRC" shutdown
Dead process did not yet have an RUJ journal	indicates that the process being recovered did not have an RUJ journal, so no recovery is necessary
Dead process transaction *number : number* did not require UNDO	indicates that the process being recovered did not require transaction undo, so no recovery is necessary
Dead process transaction *number : number* was already committed	indicates that the process being recovered committed the last transaction, so no recovery is necessary
Dead process transaction *number : number* was not ac-	indicates that the process being recovered did not

Monitor Log Message	Description
tive	have an active transaction, so no recovery is necessary
Dead process transaction *number : number* was rolled back	indicates that the process being recovered needed to have the specified transaction rolled back
for utility access	indicates that the database is being opened by a database "utility" process and may not be available for general user access
global buffer count is *number*	indicates the total number of global buffers allocated to the database
home directory "*location*" is invalid	indicates that the database monitor "home" directory is invalid or inaccessible. This also occurs if the SYS$DISK logical contains a search list when the monitor is started
ignored unknown request (*number*.) from *name*:	indicates that a strange and unexpected message was received from the specified process
image name "*name*"	identifies the image name for the user attaching to the database
image name not available	indicates that the image name for the user attaching to the database is not available
installing database dashboard	indicates that the database dashboard is being installed
lock granted to cluster recovery	indicates that this monitor will perform cluster recovery for a failed remote node's monitor
lock granted to cluster watcher	indicates that this monitor is participating in the cluster recovery mechanism
lock request completed for *name*	indicates that the specified lock was acquired
maximum global buffers per user is *number*	identifies the maximum number of global buffers available to each database user
node failure detected	indicates that a remote node failure has been detected
Not enough buffers for DBR to bind.	indicates that there are insufficient global buffers available for the database recovery process ("DBR") to recover a failed process
number of global buffers is *number*, maximum buffers	identifies the number of available global buffers and the amount that can be allocated to each database

Monitor Log Message	Description
per user is *number*	user
process name *name*, user *name*	identifies the process and user name
received *message* from *number : number*	identifies the sender of a message request
received *name* process termination from	identifies the name of a failed process
received AIJ Backup Server resume request from *number : number*	indicates that a request to resume the AIJ Backup Server ("ABS") has been received
received AIJ Backup Server start request from *number : number*	indicates that a request to start the AIJ Backup Server ("ABS") has been received
received AIJ Backup Server suspend request from *number : number*	indicates that a request temporarily suspend the AIJ Backup Server ("ABS") has been received
received AIJ Backup Server termination from	indicates that the AIJ Backup Server ("ABS") has completed
received AIJ Log Catch-Up Server log re-open request from *number : number*	indicates that a request to re-open the Log Catch-up Server ("LCS") log file has been received
received AIJ Log Catch-Up Server start request from *number : number*	indicates that a request to start the Log Catch-up Server ("LCS") has been received
received AIJ Log Catch-Up Server termination from *number*	indicates that the AIJ Backup Server ("ABS") has completed
received AIJ Log Roll-forward Server log re-open request from *number : number*	indicates that a request to re-open the Log Roll-forward Server ("LRS") log file has been received
received AIJ Log Roll-forward Server resume request from *number : number*	indicates that a request to resume the Log Roll-forward Server ("LRS") has been received
received AIJ Log Roll-forward Server start request from *number : number*	indicates that a request to start the Log Roll-forward Server ("LRS") has been received
received AIJ Log Roll-forward Server suspend request from *number : number*	indicates that a request to temporarily suspend the Log Roll-forward Server ("LRS") has been received
received AIJ Log Roll-Forward Server termination from *number*	indicates that the Log Roll-forward Server ("LRS") has completed
received AIJ Log Server log re-open request from *number : number*	indicates that a request to re-open the AIJ Log Server ("ALS") log file has been received

Monitor Log Message	Description
received AIJ Log Server start request from *number : number*	indicates that a request to start the AIJ Log Server ("ALS") log file has been received
received AIJ Log Server stop request from *number : number*	indicates that a request to stop the AIJ Log Server ("ALS") log file has been received
received AIJ Log Server termination from	indicates that the AIJ Log Server ("ALS") has completed
received AIJ Log Ship Server debug request from *number : number*	indicates that a request to debug the AIJ Log Server ("ALS") log file has been received
received AIJ Log Ship Server stop request from *number : number*	indicates that a request to stop replication of the AIJ Log Server ("ALS") log file has been received
received cluster-wide CLOSE DELPRC request	indicates that a RMU Close utility request from another node has been received, indicating processes should be terminated with DELPRC.
received cluster-wide CLOSE FORCEX request	indicates that a RMU Close utility request from another node has been received, indicating images should be terminated with FORCEX.
received cluster-wide CLOSE request	indicates that a RMU Close utility request from another node has been received, indicating processes should be shutdown normally.
received notification of remote monitor deaccess	indicates that a monitor on a remote node was shutdown normally
received Record Cache Server start request from *number : number*	indicates that a request to start the Row Cache Server ("RCS") log file has been received
received Record Cache Server stop request from *number : number*	indicates that a request to stop the Row Cache Server ("RCS") log file has been received
received Record Cache Server termination from	indicates that the Row Cache Server ("RCS") has completed
received recovery image termination from	indicates that the database recovery process ("DBR") completed
received request from remote node to join	indicates that a new monitor on a previously-unknown remote node has requested entry into the cluster recovery mechanism

Monitor Log Message	Description
received user attach request from *number : number*	indicates that a user has requested a database attach
received user image termination from	indicates that a user process has completed
record cache is *keyword*	indicates that the row cache feature is either "allowed" or "not allowed"
recovery was successful	indicates that process failure recovery was successful
released communication with process	indicates that a message buffer was released back to the "free" pool
remote monitor *number* is being recovered	indicates that the specified remote monitor is being recovered
remote monitor *number* is not responding	indicates that communications with the specified remote monitor has been interrupted
remote monitor *number* is shutdown	indicates that the specified remote monitor has been successfully shut down
remote monitor *number* is still active	indicates that the specified remote monitor is still alive and well and living in Argentina
Request has been re-scheduled due to	indicates that a message request cannot be immediately serviced and has been re-scheduled for later action
request suspended until database startup completes	indicates that a user request has been delayed until the database has been successfully opened; this is usually an attach request
sending ZZZ reply to *number : number*	indicates that a reply message is being sent to the specified user
sending image force exit to process	indicates that the specified image is being terminated with FORCEX due to database shutdown
sending process deletion to process	indicates that the specified process is being terminated with DELPRC due to database shutdown
Server name "*name*" exists	indicates that the specified server name already exists; this is a hot standby failure
shutting down ALS process	indicates that the AIJ Log Server ("ALS") process is being gracefully shutdown

Monitor Log Message	**Description**
shutting down RCS process	indicates that the Row Cache Server ("RCS") process is being gracefully shutdown
starting *keyword* shutdown of database "*name*":	indicates that database shutdown is being started
Unknown DBR status	indicates that the database recovery process ("DBR") terminated with an unknown status code
user termination suspended until recovery ready	indicates that a process failure has been suspended until the database recovery process ("DBR") is able to acquire the database freeze lock
waiting for cluster-wide close lock	indicates that the monitor is waiting to acquire the database close lock
waiting for cluster-wide DELPRC close lock	indicates that the monitor is waiting to acquire the database close/DELPRC lock
waiting for cluster-wide FORCEX close lock	indicates that the monitor is waiting to acquire the database close/FORCEX lock
XA recovery service attach request accepted	indicates that an XA-X/OPEN transaction recovery process attach request has been received and accepted

Appendix

Updating Logical Area Types

Appendix I - Updating Logical Area Type Information

Logical Area Types

Starting with Oracle Rdb 7.0.1.0, the RMU Show Statistic utility can display information on a "per-logical area". A "logical area" is a table, btree index, hash index, or any partition of one of those.

The RMU Show Statistic utility use the on-disk Area Inventory Page ("AIP") information to determine the appropriate "type" of each logical area, so that the corresponding statistics information can be displayed. The logical area "type" information in the AIP is "unknown" for logical areas created prior to Oracle Rdb 7.0.1.0. If the RMU Show Statistic utility cannot determine the logical area type from the AIP, it will prompt the user to manually enter the logical area type; however, this information is not updated in the database AIP pages.

RMU Repair Utility

Therefore, in order to update the on-disk logical area "type" in the AIP, the RMU Repair utility has been enhanced. The /INITIALIZE=LAREA_PARAMETERS qualifier option file support has been enhanced. You can specify a /TYPE=[TABLE|BTREE|HASH|SYSTEM|BLOB] in the options file. For example, to repair the EMPLOYEES table of the MF_PERSONNEL database, you would create an options file that contains the following line:

```
EMPLOYEES /TYPE=TABLE
```

For partitioned logical areas, the /AREA=*name* qualifier can be used to identify the specific storage areas that are to be updated. For example, to repair the EMPLOYEES table of the MF_PERSONNEL database for the EMPID_OVER storage area only, you would create an options file that contains the following line:

```
EMPLOYEES /AREA=EMPID_OVER /TYPE=TABLE
```

The /TYPE qualifier specifies the type of a logical area. The following keywords are allowed:

Keyword	**Description**
TABLE	Specifies that the logical area is a data table. This would be a table created using the SQL "CREATE TABLE" syntax.
BTREE	Specifies that the logical area is a btree index. This would be an index created using the SQL "CREATE INDEX TYPE IS SORTED" syntax.
HASH	Specifies that the logical area is a hash index. This would be an index created using the SQL "CREATE INDEX TYPE IS HASHED" syntax.
SYSTEM	Specifies that the logical area is a system record which is used to identify hash buckets. Users cannot explicitly create these types of logical areas. This type should not be used for the RDB$SYSTEM logical areas. This type does not identify system relations.
BLOB	Specifies that the logical area is a blob repository.

Validation

There is no error checking of the "type" specified for a logical area. The specified type does not affect the collection of statistics, nor does it affect the readying of the affected logical areas. However, an in correct type will cause incorrect statistics to be reported by the RMU Show Statistic utility.

SQL Script

The following SQL script will attempt to generate a RMU Repair utility script to update the logical area types in your database.

Note that this script may generate more entries than are actually used. This occurs because Oracle Rdb always shares the logical area between indices for the same table when mapped to RDB$SYSTEM, but the system tables do not contain that information.

Also note that you will need to edit the resulting output file and manually remove the SQL statements.

```
set noverify
set output rmu_repair.opt

-- identify the tables
select RDB$RELATION_NAME || '/TYPE=TABLE'
from RDB$RELATIONS R
where RDB$VIEW_BLR is null            -- filter views and temp tables
         and (RDB$STORAGE_ID <> 0     -- filter tables (unmapped)
              or exists (select *     -- base tables (mapped)
                         from RDB$STORAGE_MAPS
                         where RDB$RELATION_NAME = R.RDB$RELATION_NAME))
union all

-- identify the btree indices
select RDB$INDEX_NAME || '/TYPE=BTREE'
from RDB$INDICES B
where exists (select RDB$SYSTEM_FLAG  -- filter system indices
              from RDB$RELATIONS R
              where R.RDB$RELATION_NAME = B.RDB$RELATION_NAME
                and R.RDB$SYSTEM_FLAG = 0)
         and (not exists (select *          -- unmapped dflts to btree
                          from RDB$STORAGE_MAP_AREAS
                          where RDB$MAP_NAME = B.RDB$INDEX_NAME)
                    or (exists (select * -- mapped with no SYSTEM  rec
                                from RDB$STORAGE_MAP_AREAS
                                where RDB$MAP_NAME = B.RDB$INDEX_NAME
                                  and RDB$STORAGE_ID = 0))
union all

-- identify the hash indices
select RDB$INDEX_NAME || '/TYPE=HASH'
from RDB$INDICES H
where RDB$INDEX_ID = 0                        -- mapped with SYSTEM rec
  and exists (select RDB$STORAGE_ID
              from RDB$STORAGE_MAP_AREAS
              where RDB$MAP_NAME = H.RDB$INDEX_NAME
                and RDB$STORAGE_ID <> 0)
union all

-- identify the blobs
select CAST('RDB$SEGMENTED_STRINGS' AS RDB$OBJECT_NAME) || '/TYPE=BLOB'
from RDB$DATABASE
union all
```

- identify SYSTEM records

```
select CAST('RDB$SYSTEM_RECORD' AS RDB$OBJECT_NAME) || '/TYPE=SYSTEM'
from RDB$DATABASE;
```

Appendix J

Sending Stall Messages to a Mailbox

Appendix J - Sending Stall Message Information to a Mailbox

As a result of experiences with the NOTIFY and ALARM qualifiers, and with other database products, several customers have requested a feature for a future release that might make life easier for DBAs.

Using RMU/SHOW STATISTIC/NOTIFY/ALARM as an example, it is nice to have a message sent to OPCOM when the problem occurs, but should there be need to react to the information, it is nontrivial to trap the message. It would be really nice if RMU could check for the presence of a mailbox and if found write a meaning full message to it. It should not be too difficult to extend this to other events also, thus giving DBAs a simple interface to assist in automating the system and of automating recovery reaction to problems.

The purpose of this section is to describe one method for easily achieving the above request using the RMU Show Statistic utility. Starting with Oracle Rdb 7.0.0.0, the RMU Show Statistic utility contains a STALL_LOG=*filespec* qualifier. The purpose of this qualifier is to specify the filespec of a log file to which stall messages are sent. The specified filespec can be a disk file, a terminal or a OpenVMS mailbox.

While directing the stall messages output to a mailbox using the STALL_LOG qualifier is not very difficult, the hard part is actually creating the OpenVMS mailbox; there is no command to create a mailbox from DCL, so a small user-written program is needed.

The C language source code for a program to create a mailbox is provided at the end of this section. The source code will need to be compiled and linked into an executable; the resulting executable will then have to be defined as a DCL foreign command before it can be used.

The following example demonstrates how to do this.

```
$ CC MAILBOX.C /STAND=VAXC
$ DEFINE LNK$LIBRARY SYS$LIBRARY:VAXCRTL
$ LINK MAILBOX.OBJ
$ MAILBOX :== $DISK:[DIR]MAILBOX.EXE
```

The following example describes the steps needed to actually create the mailbox, using the user-written program, and how to use the STALL_LOG qualifier to identify the mailbox.

```
$ MAILBOX -c DEMO
Mailbox "demo" successfully created using device "_MBA3677:"
$ RMU Show Statistic /STALL_LOG=DEMO MF_PERSONNEL
```

Note that the PRMMBX privilege is required to create the permanent mailbox. The SYSNAM privilege is required to create the system logical.

The SYSNAM privilege requirement can be avoided if the mailbox is created without a name. The following example shows how to do this.

```
$ MAILBOX -c
Mailbox successfully created using device "_MBA3679:"
$ RMU Show Statistic /STALL_LOG=_MBA3679: MF_PERSONNEL
```

The database stall messages will now be directed to the newly created mailbox. Note that the RMU Show Statistic utility sending the stall messages to the mailbox will hang if the mailbox is not emptied fast enough by another process. The following example shows one method that can be used to access the contents of the mailbox by another process; note that there are many other methods that can be employed to access the mailbox.

```
$ COPY DEMO TT:
```

The result of the above example will be the stall messages displayed on the process' terminal, as the output is being produced by the database process.

The process that reads from an unnamed mailbox must access the mailbox by its device name. The device name was displayed when the mailbox was created. The following example shows how to read from an unnamed mailbox.

```
$ COPY _MBA3679: TT:
```

Remember that the mailbox must be explicitly deleted when it is no longer needed. The following example shows how to delete the mailbox.

```
$ MAILBOX -d DEMO
Mailbox "demo" successfully deleted (device "_MBA3677:")
```

The following is the "C" source code to the MAILBOX utility.

```
/* MAILBOX.C Original: Feb 1990 Albert Godfrind */
/* Last Modified: Jan 1998 Rick Anderson (rjanders@us.oracle.com) */
#include <STDIO.H>
#include <DESCRIP.H>
#include <SSDEF.H>
#include <DVIDEF.H>
#define TRUE 1
#define FALSE 0
extern void mailbox_create();
extern void mailbox_delete();
extern void exit();
extern char * strcpy();
extern int strlen();
extern long sys$crembx();
extern long sys$delmbx();
extern long sys$getdviw();
extern long sys$assign();

main(argc, argv)
int argc; char * argv[];
{
int argno; /* command-line argument index */
char * arg; /* --> command-line argument */
int prt_usage; /* print usage message? flag */

prt_usage = FALSE;
for ( argno = 1; argno < argc; ++argno )
  {
  arg = argv[ argno ];
  if ( *arg == '-' )
    {
    switch ( *(++arg) )
      {
      case 'c': /* create a mailbox */
      {
      /* mailbox name is optional */
      if ( (argno + 1) < argc )
        {
        mailbox_create(argv[ ++argno ]);
        }
      else
        {
        mailbox_create("");
        } /* endif */
      break;
```

```
      } /* endcase */
    case 'd': /* delete a mailbox */
      {
      /* mailbox name is mandatory */
      if ( ++argno >= argc )
        {
        fprintf(stderr, "mbx: Mailbox name not specified\n");
        prt_usage = TRUE;
        break;
        } /* endif */
      mailbox_delete(argv[ argno ]);
      break;
      } /* endcase */
   default: /* unknown command-line option */
      {
      fprintf(stderr, "mbx: Unknown option '-%s'\n", arg);
      prt_usage = TRUE;
      } /* endcase */
    } /* endswitch */
  }
else
  {
  fprintf(stderr, "mbx: Unknown option '%s'\n", arg);
  prt_usage = TRUE;
  } /* endif */
} /* nextfor */

if ( (prt_usage) || (argc == 1) )
  {
  fprintf(stderr, "usage: mbx -c [MAILBOX_NAME]\n");
  fprintf(stderr, " or: mbx -d MAILBOX_NAME\n");
  exit(1);
  }
exit(1);
}

void mailbox_create(name)
char * name;
{
int mbx_channel; /* mailbox channel number */
long return_status;
char mbx_name[512];
char mbx_dev_name[64];
int mbx_dev_name_l;
struct item_list_d
  {
  unsigned short int size;
```

```
  unsigned short int code;
  unsigned long int address;
  unsigned long int ret_length;
  unsigned short int size_l;
  unsigned short int code_l;
  } item_list;

$DESCRIPTOR (mbx_name_dsc, mbx_name);
if ( *name != '\0' )
  {
  strcpy(mbx_name, name);
  mbx_name_dsc.dsc$w_length = strlen(mbx_name);
  return_status = sys$crembx(1, &mbx_channel, 0, 0, 0, 0, &mbx_name_dsc);
  }
else
  {
  return_status = sys$crembx (1, &mbx_channel, 0, 0, 0, 0, 0);
  } /* endif */
if ( return_status != SS$_NORMAL )
  {
  fprintf(stderr, "mbx: Unable to create mailbox\n");
  exit(return_status);
  } /* endif */

item_list.size = 64;
item_list.code = DVI$_DEVNAM;
item_list.address = (char *)&mbx_dev_name;
item_list.ret_length = (long *)&mbx_dev_name_l;
item_list.size_l = 0;
item_list.code_l = 0;
return_status = sys$getdviw(0, mbx_channel, 0, &item_list, 0, 0, 0, 0);
if ( return_status != SS$_NORMAL )
  {
  fprintf(stderr, "mbx: Unable to acquire mailbox information\n");
  exit(return_status);
  } /* endif */

if ( *name != '\0' )
  {
  printf("Mailbox \"%s\" successfully created using device \"%s\"\n",
    mbx_name, mbx_dev_name);
  }
else
  {
  printf("Mailbox successfully created using device \"%s\"\n",
   mbx_dev_name);
  } /* endif */
```

```
return;
}

void mailbox_delete(name)
char * name;
{
int mbx_channel; /* mailbox channel number */
long return_status;
char mbx_name[512];
char mbx_dev_name[64];
int mbx_dev_name_l;
struct item_list_d
  {
  unsigned short int size;
  unsigned short int code;
  unsigned long int address;
  unsigned long int ret_length;
  unsigned short int size_l;
  unsigned short int code_l;
  } item_list;

$DESCRIPTOR (mbx_name_dsc, mbx_name);
strcpy(mbx_name, name);
mbx_name_dsc.dsc$w_length = strlen(mbx_name);
return_status = sys$assign (&mbx_name_dsc, &mbx_channel, 0, 0);
if ( return_status != SS$_NORMAL )
  {
  fprintf(stderr, "mbx: Unable to access mailbox\n");
  exit(return_status);
  } /* endif */
item_list.size = 64;
item_list.code = DVI$_DEVNAM;
item_list.address = (char *)&mbx_dev_name;
item_list.ret_length = (long *)&mbx_dev_name_l;
item_list.size_l = 0;
item_list.code_l = 0;
return_status = sys$getdviw(0, mbx_channel, 0, &item_list, 0, 0, 0, 0);
if ( return_status != SS$_NORMAL )
  {
  fprintf(stderr, "mbx: Unable to acquire mailbox information\n");
  exit(return_status);
  } /* endif */
return_status = sys$delmbx(mbx_channel);
if ( return_status != SS$_NORMAL )
  {
  fprintf(stderr, "mbx: Unable to delete mailbox\n");
  exit(return_status);
```

```
  } /* endif */
printf("Mailbox \"%s\" successfully deleted (device \"%s\")\n",
  mbx_name, mbx_dev_name);
return;
}
```

Appendix K

Real-World Database Tuning Tips

Appendix K - Real-World Database Tuning Tips

This section contains real-world database tuning tips, obtained from customers and consultants solving performance problems on production databases. All of the information provided in this section is provided by customers of the Oracle Rdb product.

Hopefully, the tuning tips described apply to your particular database performance issues as well.

Average I/O Stall Time

One of the first things to do is to compare the average IO stall time per transaction, the total of the average lock stall times per transaction and the average transaction duration.

This gives the proportions of an "average" transaction that are spent waiting for I/O and for locks. which indicates the maximum gain from reducing the stalls. For example, if about 20% of the time is spent waiting for locks then reducing the lock stalls to zero will give a possible 25% increase in throughput, but if 60% is waiting for I/O to complete then that allows for a possible 150% increase, so I should concentrate on the I/O rather than the locks.

Excessive SPAM Page Access

High SPAM page fetch rate and low number of pages-discarded are usually symptoms of a new database attach (process) trying to establish a target page for a very large uniform format storage area. The algorithm for selecting a target page is something like:

1. Use the last page on which a record was successfully erased or stored.

2. If no such page, search all pages in current buffers (marked pages first, then unmarked)

3. Use first page of area, scanning SPAM pages until a suitable page is found.

If the logical area is very large and densely packed, then the first attempt to insert into that logical area will cause a long search of the SPAM pages.

The following is a work-around to this problem:

1. Look up a row that was most recently stored or otherwise is most likely to be at the end of the storage area to establish the target page.

2. Alter storage map periodically so that all new records get stored in an empty storage area.

Excessive Pages Checked/Discarded

In the "Record Analysis" screen, the following is displayed:

```
66.9% page discard rate above 10.0% threshold (avg 0.3 I/Os)
100.0% SPAM page fetch rate above 80.0% total fetched threshold
401439.7% SPAM page fetch rate above 80.0% record stored threshold
```

However, the buffer pool effectiveness is 98%. There is no fragmentation in the database. The RMU/ANALYZE/AREAS/OPTION=FULL utility on the most highly used areas shows nothing significant.

To determine which logical area (table or index) is causing the reported problem, try the following:

1. Be sure you are running the RMU Show Statistic utility with the /LOGICAL_AREA qualifier.

2. Dump the "full" numbers report. This is accomplished by selecting the "Options" on-screen menu, then choosing the "Numbers" option.

3. From the OpenVMS command line, issue the following DCL commands:

```
$ search statistics.rpt discarded/out=a.a
$ search a.a 1,2,3,4,5,6,7,8,9/out=b.b
```

4. Edit the file b.b and note down the numbers.

5. Edit the statistics.rpt file and find the numbers you've noted. The number will be in the logical area screen pertaining to the table and/or index that has the problem.

Based on my experience, you can be reasonably certain that you have one or more indexes with duplicates allowed without any threshold defined. Look into *Guide to Database Maintenance and Performance* on how to calculate logical area thresholds, rebuild the indexes and you are done.

The second SEARCH command is aimed to exclude the rows where the counters are all zeros. This is meaningful in case you have many tables and/or indexes defined. Use this procedure every now and then to verify that everything is okay as far as the "pages discarded" problems are concerned.

Excessive I/O to RDB$SYSTEM

A database is experiencing 75% of the total I/O operations to the RDB$SYSTEM storage area. Examining the RMU Extract utility output, 41 tables and 10 indices are stored in the RDB$SYSTEM area. However, the these are static tables that are not being used. Most of the I/O to RDB$SYSTEM is synchronous reads. If the database is using global buffers, the I/O to the RDB$SYSTEM storage area decreased a lot, indicating their users are re-reading the same pages quite a bit.

The logical area statistics were designed with these types of problems in mind. If the customer is using an Oracle Rdb version prior to Rdb7, then the next best thing is to check the stall screens (crank the refresh rate down) and see what pages are being stalled for either reads or writes in physical area 1 and then dump the pages to see what is on them.

The first recommendation is to move the tables and indices out of RDB$SYSTEM. Try checking locks to make sure that the tables that are not being used do not have locks on them. Another possibility is that the records in those tables are badly fragmented and for each of one of the I/Os you expect you get a large number of I/O operations, all synchronous.

It is also possible that the RDB$FIELD_VERSIONS table has lots of entries, indicating that metadata updates have been done against tables but not all rows know about the changes?

It might also be interesting to get the cardinalities for the system tables.

If the customer is using a lot of read-only transactions, then this is a known bottleneck. Oracle Rdb must read the AIP pages during the SET TRANSACTION statement to ensure the referenced areas were not updated by an EXCLUSIVE transaction. In Oracle Rdb7 you can change the RDB$SYSTEM storage area to be read-only to eliminate the I/O; the AIP information is cached in memory in this case. Oracle Rdb release 7.0.1.3 changed the way this information was checked and eliminated most of the I/O associated with this function.

Excessive SEQBLK Stalls

Starting with Oracle Rdb7, the most visible stall on a database is the SEQBLK lock. This lock is necessary for the correct function of Oracle Rdb. It controls the allocation of the unique transaction number required for each user's transaction (TSN). The lock is extremely fast but becomes stalled by read-only transactions, not read/write transaction as one would assume.

Therefore, we must find ways to increase the speed of read-only transactions and find ways to increase the speed of the database rootfile.

The SEQBLK lock existed in V6, but it was included in a single generic lock, RWROOT, and was not as visible due to other more-frequent stall points in the product. In Oracle Rdb7, SEQBLK lock is separated from a single RWROOT lock into a series of separate locks, to improve contention for a single resource.

Oracle Rdb7 is true 64-bit processing product and the size of the TSN has increased from 32 to 64 bits. In Oracle Rdb v6.1 and earlier, one TSN block could contain 50 users. However, starting with Oracle Rdb7, one TSN block can contain only 28 users. This means that twice as many TSN blocks are needed for the same number of users. In addition, since Oracle Rdb7 is considerably faster than Oracle Rdb v6.1, the transactions complete faster and experience significantly fewer stalls. Therefore, it is possible to encounter many more SEQBLK lock stalls than in prior versions. The TSN blocks are stored in the database rootfile.

To improve the updating of the TSN blocks, it is necessary to make access to the database rootfile faster using physical methods. Some ideas include the following :

- Move the database rootfile to separate disk device, with little competition from other processes.
- If possible, have a dedicated device controller for the database rootfile disk device.
- If possible, use write-back caching on the rootfile device controller.
- Shadow the database rootfile disk to improve read access to TSN blocks. However, it will slow down the update, so this idea may not help if the transaction frequency is extremely high.
- Consider using the Commit to Journal Optimization feature, if possible.
- Move the database rootfile to a solid state disk. A solid state disk is similar to memory, but can be shared by each member in the cluster. A small solid state disk can perform 4000-5000 I/Os per block per second. A TSN Block is 512 Bytes (1 block), therefore, this is an excellent use of a solid state disk. A solid state disk is expensive, but is effective for this purpose.
- Reduce the number of users that access the database to be closer to the actual number of users. There will be fewer TSN Blocks to search.

Since the major cause of blocking of TSN processes are the read-only transactions, we must find ways to make the read-only access faster. Some ideas include the following :

- Use volume shadowing on the storage areas that have many read-only transactions.
- If the database is replicated, possibly by Hot Standby, ODBC users could attach to the replicated database. I do not recommend installing Hot Standby until V7.0.2 for this environment.
- Eliminate read-only transactions. As an alternative, consider using "Read Write Reserving .... for Shared Read". When a application changes its transaction mode from read-only to

read/write, the buffers are flushed and a pre-started transaction is wasted. Instead, if the program begins with a read/write transaction (reserving ... shared read) and changes to "reserving ... shared write", the buffers are not flushed and the pre-started transaction is used. This method should reduce locking the TSN block updates and reduce or eliminate stalls by SEQBLK locks. This method is the current practice in many of the applications. It is a simple change but potentially affects many programs.

- When read-only transactions are eliminated, there are other database changes that may be effective in this environment, such as "commit to journal optimization and deferring snapshots.

Glossary

Glossary

2PC. A common abbreviation for "Two Phase Commit", a distributed transaction commit protocol.

2PL. A common abbreviation for "Two Phase Locking", a locking protocol in which locks are all acquired and then all released.

Abort. The act of terminating a transaction and atomically undoing all of its actions.

ABS. An abbreviation for "AIJ Backup Server". A background process invoked by the database monitor to automatically backup after-image journals.

ABW. An abbreviation for "Asynchronous Batch Write". An Oracle Rdb feature whereby modified database buffers are lazily written back to the database asynchronously.

APF. An abbreviation for "Asynchronous Pre-Fetch". An Oracle Rdb feature whereby database pages are fetch asynchronously in advance of actually being needed. This feature is limited to sequential scans.

ADT. A common abbreviation for "Abstract Data Type", typically used in conjunction with object-oriented attributes. Not to be confused with database rootfile "objects".

After-Image. A log record containing a record image after a modification has been made to it. See also "Before-Image".

After-Image Journal. A database journal with a ".AIJ" file type, containing after-images of record modifications. See also "Recovery-Unit Journal".

AIJSERVER. A network object process used to consume Hot Standby network messages from the master database and transfer them to the LRS process to be applied to the standby database.

ALS. An abbreviation for "AIJ Log Server". A background process that serves as the after-image journal group commit process.

ARB. An abbreviation for "AIJ Request Block". Page modifications to be written to the after-image journal are packaged into ARBs and queued for submission.

Atomic Action. The basic principle of the transaction, in which all actions either complete or appear as if nothing was ever attempted.

Availability. A measure of scheduled time that the database or application provides service.

B-Tree Index. Also known as a "btree" index, a database access method that provides keyed search capabilities, both direct and sequential. See also "Hash Index".

Before-Image. A log record containing a record image before a modification has been made to it. See also "After-Image".

Checkpoint. The moment in time when a transaction is fully committed to the database. This term applies when using the "Fast Commit" feature. All transaction modifications after this point in time may have to be re-done in case of process or database failure.

Cluster. A collection of computers that acts as a single unit. Within a cluster, all nodes can access all devices but do not share physical memory.

Commit. The act of making a transaction's actions durable and visible to other database users.

Configuration File. A human-readable file containing RMU Show Statistic utility parameters, attributes and preferences.

CSN. An abbreviation for "Commit Sequence Number". Uniquely identifies a transaction. Identifies the order in which transactions commit.

DAPF. An abbreviation for "Detected Asynchronous Pre-Fetch". An Oracle Rdb feature whereby database pages are fetch asynchronously in advance of actually being needed, based on detectable access patterns. This feature is not limited to sequential scans.

DBA. An abbreviation for "Database Administrator". The person responsible for database maintenance and performance analysis using the RMU Show Statistic utility.

Deadlock. A situation in which two or more transactions are each stalled on an object owned by one of the other transactions. The transactions will remained stalled indefinitely until one of the transactions aborts.

DECnet. A network protocol used by OpenVMS.

Disaster Recovery. The ability to quickly recover from a catastrophic loss of an entire database, typically achieved by using a Hot Standby database.

Distributed Transaction. A transaction involving two or more transaction managers or databases. See also "2PC".

Double Point-of-Failure. A secondary database failure following a single point-of-failure that the database is not capable of recovering.

Fast Commit. A performance mechanism whereby page buffers do not have to be flushed to disk at transaction commit.

Group Commit. A mechanism whereby multiple committing transactions are written to the after-image journal using a single I/O operation, thereby amortizing the I/O expense across all committing transactions.

Hash Index. A database access method that maps a key via a hashing algorithm uniformly across a partitioned set of database pages. Cannot be used for sequential scans. See also "B-Tree Index".

Hot Standby. A separate database maintained as a "copy" of a master database available in case disaster recovery is needed.

LCS. An abbreviation for "Log Catch-up Server". A background process on a master database whose job it is to synchronize the database with the standby database.

Logical Area. A table, btree index, hash index, blob or system record. Logical areas reside in physical areas.

Long Stream. A database operation typically performed in the database kernel and not interruptible.

LRS. An abbreviation for "Log Replication Server". A background process on the standby database whose job it is to automatically apply AIJ information received from the master database.

MSN. An abbreviation for "Message Sequence Number".

PID. An abbreviation for "Process Identifier". Uniquely identifies a process attached to a database. See also "TID".

Physical Area. A storage area file (".RDA") within which logical areas reside.

RAID. An abbreviation for "Redundant Arrays of Independent Disks". A disk device consisting of multiple disks capable of providing error-correcting characteristics.

Recovery-Unit Journal. A database journal with a ".RUJ" file type, containing before-images of record modifications. See also "After-Image Journal".

Rootfile. A database file with a ".RDB" file type, containing the database control structures known as "objects".

Scalability. A measure of how well the database or application is able to adapt to future business needs.

Single Point-of-Failure. A single database failure event that the database is capable of recovering.

SMP. An abbreviation for "Symmetric Multi-processor". A computer comprised of two or more CPUs capable to executing multiple tasks simultaneously.

STID. An abbreviation for "Stream Identifier". Uniquely identifies a process thread.

Storage Area. A database file with a ".RDA" file type. See also "Physical Area".

Switch-Over. The act of moving transaction logging from one after-image journal to another, typically when the first fills up.

Synthesized Page. A requested page that is synthesized is either read from a WORM storage area or read from a uniform format area and the clump is unallocated.

TCP/IP. A network protocol commonly used by most non-OpenVMS platforms, such as UNIX. Also available on OpenVMS.

TID. An abbreviation for "Transaction Identifier". Uniquely identifies a process attach to a database. See also "PID".

Timeout. A threshold placed on a resource acquisition request. When the request duration exceeds the threshold, the request is canceled.

TPS. An abbreviation for "Transactions Per Second". A measure of transactional through-put. This term is often used in conjunction with standardized benchmarks, but is used by the RMU Show Statistic utility as an accurate measurement of the actual database through-put.

Transaction. An atomic unit of database work.

TSN. An abbreviation for "Transaction Sequence Number". Uniquely identifies a transaction. Identifies the order in which transactions start.

Verb. Typically, a SQL statement, or block of statements, that must be executed atomically.

WORM. An abbreviation for "Write Once Read Many". An optical disk with the properties that each block may only be written once and never again. However, a WORM device maintains a consistent view of a logical block number regardless of the number of times the block has been modified.

INDEX